NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

A GENERALIZED ORIENTEERING PROBLEM FOR OPTIMAL SEARCH AND INTERDICTION PLANNING

by

Jesse Pietz

September 2013

Dissertation Supervisor: Johannes O. Royset

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
23-07-2013	Dissertation	01-02-2011 – 23-07-2013

4. TITLE AND SUBTITLE

A GENERALIZED ORIENTEERING PROBLEM FOR OPTIMAL SEARCH AND INTERDICTION PLANNING

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Jesse Pietz

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government. IRB Protocol Number: N.A.

14. ABSTRACT

In order to support search planning for counterdrug operations, we introduce a generalized Orienteering Problem (OP) where transit on arcs in a network and reward collection at nodes both consume a variable amount of the same limited resource. We exploit this resource trade-off through a specialized branch-and-bound algorithm that relies on partial path relaxation problems, which often yield tight bounds and lead to substantial pruning in the enumeration tree. We present the Smuggler Search Problem (SSP) as a real-world application of our generalized OP. Numerical results show that our algorithm applied to the SSP outperforms standard mixed-integer nonlinear programming solvers for problems with seven or more targets. We present model enhancements that allow practitioners to represent realistic search planning scenarios. We investigate how evolving uncertainty in planning data can be addressed by a multi-stage stochastic programming model.

15. SUBJECT TERMS

Counterdrug operations, Mixed-integer nonlinear programming, Optimal search, Orienteering problem, Search and interdiction, Search theory, Smuggler search problem, Vehicle routing problem

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
Unclassified	Unclassified	Unclassified	UU	161	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

A GENERALIZED ORIENTEERING PROBLEM FOR OPTIMAL SEARCH AND INTERDICTION PLANNING

Jesse Pietz
Major, United States Air Force
B.S., United States Air Force Academy, 2001
M.A., Rice University, 2003

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 2013

Author: Jesse Pietz

Approved by:

Johannes O. Royset
Associate Professor of
Operations Research
Dissertation Supervisor

W. Matthew Carlyle
Professor of
Operations Research

Ronald D. Fricker, Jr.
Professor of
Operations Research

Craig W. Rasmussen
Professor of
Applied Mathematics

Michael P. Atkinson
Assistant Professor of
Operations Research

Approved by: Robert F. Dell
Chair, Department of Operations Research

Approved by: O. Douglas Moses
Vice Provost for Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In order to support search planning for counterdrug operations, we introduce a generalized Orienteering Problem (OP) where transit on arcs in a network and reward collection at nodes both consume a variable amount of the same limited resource. We exploit this resource trade-off through a specialized branch-and-bound algorithm that relies on partial path relaxation problems, which often yield tight bounds and lead to substantial pruning in the enumeration tree. We present the Smuggler Search Problem (SSP) as a real-world application of our generalized OP. Numerical results show that our algorithm applied to the SSP outperforms standard mixed-integer nonlinear programming solvers for problems with seven or more targets. We present model enhancements that allow practitioners to represent realistic search planning scenarios. We investigate how evolving uncertainty in planning data can be addressed by a multi-stage stochastic programming model.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AOI Area of Interest
B&B Branch-and-Bound
BARON Branch-And-Reduce Optimization Navigator
BONMIN Basic Open-source Nonlinear Mixed Integer programming
CS Cyclic Seesaw
DICOPT DIscrete and Continuous OPTimizer
DP Dynamic Programming
EV Expected Value
GOP-RDR Generalized Orienteering Problem with Resource Dependent Rewards
ISR Intelligence, Surveillance, and Reconnaissance
JIATFS Joint Interagency Task Force South
KKT Karush Kuhn Tucker
MCP Maximum Collection Problem
MINOS Modular In-core Nonlinear Optimization System
MIP Mixed-Integer Program
MINLP Mixed-Integer NonLinear Program
NLP NonLinear Program
NPS Naval Postgraduate School
NP Non-deterministic Polynomial-time
OP Orienteering Problem
OPVP Orienteering Problem with Variable Profits
PBB Partial Branch-and-Bound
PH Progressive Hedging
SAP Search Allocation Problem
SNOPT Sparse Nonlinear Optimizer
SSP Smuggler Search Problem
STSP Selective Traveling Salesman Problem
SVRP Selective Vehicle Routing Problem
TOP Team Orienteering Problem
VRP Vehicle Routing Problem
VRPSP Vehicle Routing Problem with Selective Pickups

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

This research is motived by the ongoing efforts of the Joint Interagency Task Force South (JIATFS), which conducts search operations in order to stem the flow of illicit traffickers from South and Central America. Planning real-world search operations is a particularly difficult task. Planners, operating under strict time constraints, weigh uncertain information about target whereabouts against the limitations of their search assets in order to develop a search plan that they expect will produce a "good" outcome. The problem of routing search assets in order to *maximize* the value of the plan is even more difficult. Path-constrained optimal search problems, such as this one, are known to be NP-hard. There are no known algorithms that can solve these problems in polynomial time. Thus, the practical difficulty associated with real-world search planning is compounded by the technical difficulty of an NP-hard optimization problem. Aiding the search planning process requires models and algorithms that give planners the ability to both capture the important features of real-world scenarios and solve problems in a reasonable amount of time.

In order to aid the search planning process, we generalize the Orienteering Problem (OP), where a vehicle is routed from a prescribed start node, through a directed network, to a prescribed destination node, collecting rewards at each node visited, in order to maximize the total reward along the path. In our generalization, transit on arcs in the network and reward collection at nodes both consume a variable amount of the same limited resource. As such, we name this problem the Generalized Orienteering Problem with Resource Dependent Rewards (GOP-RDR). We exploit the resource trade-off through a specialized branch-and-bound algorithm (B&B) that relies on partial path relaxation problems, which often yield tight bounds and lead to substantial pruning in the enumeration tree. The GOP-RDR and B&B can be used to model and solve problems in many application areas, such as mission planning for search aircraft, commercial vehicle routing, sports, tourism, production, and scheduling.

We present the Smuggler Search Problem (SSP), a novel path-constrained optimal search model in continuous time and space, as an important special case of the GOP-RDR. The SSP is used to find an optimal search plan for a single searcher that is routed in an area of interest (AOI) to detect multiple linearly moving targets. Many practical problems involve planning for 10 or fewer targets. Numerical results show that B&B applied to randomly

generated SSP instances with seven or more targets outperforms standard mixed-integer nonlinear programming solvers. We present a search order heuristic that can be used to quickly compute good search plans, with runtimes within one second for problems of up to 10 targets. We demonstrate that the search plan values of solutions computed by the search order heuristic are within 1-3% of the optimal search plan values for problem instances with up to 10 targets.

We develop five enhancements to the SSP, along with tailored solution procedures, that can be used together or separately to model various real-world search planning scenarios. The merits of each SSP enhancement are demonstrated on a benchmark scenario, which is developed to highlight many of the issues that may be faced by real-world search planners.

In the first SSP enhancement, we account for complex target motion by approximating each target's movement path with a piecewise-linear track of segments. This model may be used in planning scenarios where the target's speed and/or direction changes along its path, as well as when the searcher's performance characteristics vary in the AOI (e.g., due to environmental conditions). We demonstrate that this model, with seven target segments and a single searcher, can be solved to optimality using B&B in 3.5 seconds.

In the second SSP enhancement, we account for multiple cooperating, heterogeneous searchers. We solve problem instances of this type using B&B on a searcher-expanded network. Many practical problems involve planning for one or two searchers. We demonstrate that this model, with seven target segments and two searchers, can be solved to optimality using B&B in 48.9 seconds, while heuristic solutions, with search plan values within 10% of the optimal plan, can be computed in as little as 1.3 seconds.

With the third SSP enhancement, we demonstrate that scenarios with high uncertainty in a target's movement track may be modeled by allowing search in fixed regions. This model allows planners to account for targets whose departure time uncertainty value is larger than the duration of the planning horizon. We demonstrate that this model, with eight target segments and two searchers, can be solved to optimality using B&B in 64.8 seconds, while heuristic solutions, with search plan values within 10% of the optimal plan, can be computed in as little as 1.4 seconds.

In the fourth SSP enhancement, we coordinate aerial search efforts with the positioning of surface interdictors. This is an important consideration in counterdrug operations

where searchers must rely on surface assets to physically intercept smugglers once they are detected by the searcher. By adding continuous variables and convex constraints to the search model, we develop the first search-and-interdiction model that can be used for planning in real-world counterdrug operations. We demonstrate that this model, with eight target segments, two searchers, and four interdictors, can be solved to optimality using B&B in 84.8 seconds, while heuristic solutions, with search plan values within 10% of the optimal plan, can be computed in as little as 1.5 seconds.

The fifth SSP enhancement is a multi-period search model that accounts for sequencing search plans over multiple mission execution cycles. We show that this multi-period model may be used to improve plans when search needs to be coordinated with the positioning of interdictors over several planning periods. We demonstrate that this model, with 16 target segments, two searchers, four interdictors, and three mission execution cycles, can be solved to optimality using B&B in 6.9 hours. Runtimes as short as 3.1 seconds can be achieved by heuristics, which yield solutions with search plan values that within 19% of the optimal plan.

We study the sensitivity of optimal search plans with respect to environmental uncertainty. We demonstrate that planners need to consider the certainty level associated with scenario data, be aware of how changes to this data may affect the plan, and, if possible, directly account for data uncertainty by using a multi-stage search planning model.

We present a multi-stage stochastic programming model that allows planners to account for evolving uncertainty by considering a scenario-based planning approach. We show that a scenario-based approach, as opposed to planning for a single scenario, can yield more valuable search plans with lower risk of poor performance in worst-case scenarios. We demonstrate that the multi-stage model, with five target segments, a single searcher, and four scenarios, can be solved to optimality using B&B in 1.5 hours. In order to solve larger problem instances, we show that the progressive hedging algorithm may be used to compute solutions to multi-stage problems with many scenarios. We demonstrate that the multi-stage model, with five target segments, a single searcher, and up to 100 scenarios, can be solved using progressive hedging in within 1.5 hours.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgements

It is not possible to adequately express my gratitude to the host of people who have played a part in getting me to this point. The following is a short list; to those whose names I omitted, I offer both an apology (for my oversight) and a thank you (for your help).

I would like to thank my advisor, Professor Johannes Royset, for his mentorship, guidance, and expertise throughout this process. His calm, steady approach to navigating the vast unknown that is academic research was exactly what I needed. For their support and advice along the way, I would like to thank my committee members, Professors Mike Atkinson, Matt Carlyle, Ron Fricker, and Craig Rasmussen. I would also like to thank Professors Jerry Brown, Moshe Kress, and Al Washburn, who generously offered their time and expertise despite that fact that they were not on my committee. I thank the NPS OR faculty for their top-notch teaching and advice.

I would also like to express my gratitude to the Air Force and the Air Force Academy for giving me the opportunity to pursue a doctoral degree. I would like to thank Major General Kane, Mr. Merrill, Mr. Lindner, Lt Col Sriver, Dr. Bedrossian, and Dr. Heinkenschloss for their mentorship and support throughout my professional and academic career.

To my office-mates, Jay, Christian, Dick, Sofia, Matt, and Gary, thanks for your friendship and assistance over the last couple of years. I wish you all great success and the best of luck!

For their love and understanding, I owe a "ginormous" debt of gratitude to my kiddos. To my loving wife Adriana, thank you for your patience and your unwavering support.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

1.1 Motivation and Background

This research is motived by the ongoing efforts of the Joint Interagency Task Force South (JIATFS) to stem the flow of illicit traffickers (a.k.a., smugglers or targets) from South and Central America. This mission is large and complex. JIATFS is tasked to not only search for and detect illicit traffickers, but it must also plan to monitor them and facilitate their interdiction once detected (Joint Interagency Task Force South 2013). We consider a search planning problem where aerial searchers are routed within an area of interest (AOI) to detect and monitor targets, while surface interdictors are assigned to locations in the AOI so that targets can be interdicted when a detection occurs.

Current search planning is done manually by planners with years of operational experience. These planners weigh a number of variables associated with targets (e.g., time, space, uncertainty, and priority) against the constraints on their search and interdiction assets (e.g., availability, capability, location, speed, and endurance) in order to develop a coordinated search plan that they expect will result in seizing a large amount of illicit material. In order to aid the JIATFS search planning process, we introduce the Generalized Orienteering Problem with Resource Dependent Rewards (GOP-RDR), we develop efficient algorithms to solve instances of the problem, and we explore model extensions that give planners the ability to solve realistic planning problems.

The GOP-RDR seeks to route a vehicle along a simple path through a directed network, between prescribed start and end nodes, in order to maximize the total reward along the path. Rewards are collected by the vehicle at each node visited, where the reward level depends on the amount of scarce resources expended. Arcs in the network are traversed while consuming the same limited resources used for reward collection. Resource consumption at nodes and on arcs depend on nonlinear functions which are defined on the network. The path is constructed so that the total resource expenditure is within given limits. The GOP-RDR is a generalization of the well-known Orienteering Problem (OP), as well as several other related optimization problems: the Selective Vehicle Routing Problem (SVRP), the Selective Traveling Salesman Problem (STSP), and the Maximum

Collection Problem (MCP). In these problems, node visitation rewards and arc traversal resource expenditures (arc lengths) are fixed quantities.

The GOP-RDR arises in military, search and rescue, and law enforcement operations where the objective is to plan a searcher route to find moving targets in an AOI so that the total reward garnered by the search is maximized. In these optimal search problems, targets can be thought of as carrying some type of material that is valuable to the searcher. Thus, the reward garnered by the search is related to the amount of material detected. Limited resources (e.g., time and fuel) are expended by the searcher while performing search actions in regions of interest (the nodes in our directed network) and while in transit between these regions (the arcs in our directed network) as targets move in the AOI. The GOP-RDR may also arise in commercial applications where a vehicle can be routed to a number of locations in order to perform a service. The distance between locations can be represented by a travel time, possibly changing with time-of-day or environmental effects, and the reward garnered by performing the service at each location may be an increasing function of time spent at the location (see, for example, Yi 2003). The distinguishing characteristic of the GOP-RDR is that there exists a trade-off between resource usage in transit between the nodes and resource usage collecting rewards at the nodes.

We introduce the Smuggler Search Problem (SSP), a path-constrained optimal search problem in continuous time and space as an important example of the GOP-RDR. The SSP deals with the high level decision of routing search vehicles through subsets, *search regions*, of the AOI in the presence of uncertain information about target whereabouts.

1.2 Literature Review

The OP has received much attention in the literature; see, e.g., Vansteenwegen et al. (2011) for a recent survey. The OP has wide-ranging applicability and has been used to solve many practical problems. Tsiligirides (1984) describes the sport of orienteering, where competitors travel to various control locations to receive points as reward. The player who accumulates the most total points within the prescribed time limit is declared the winner. See also Butt and Cavalier (1994) and Golden et al. (1987) for other sporting-related OP formulations. Another common OP application is found in the tourism industry (Wang et al. 2008), where tourists visiting a country would like to plan their visit in such a way that they maximize the value of their trip. See also Silberholz and Golden (2010),

Schilde et al. (2009), and Souffriau et al. (2008) for other tourism related OP applications. The OP is used to obtain optimal mission plans for military intelligence, surveillance and reconnaissance (ISR) aircraft. Moser (1990) uses a multi-vehicle generalization of the OP to compute optimal plans. Royset and Reber (2009) consider a more general problem by adding considerations for aircraft take-off times, airspace deconfliction, and distinguishing between search and transit. OP applications related to commercial service and vehicle routing are commonly found in the literature (see, e.g., Tricoire et al. 2010, Tang and Miller-Hooks 2005, Golden et al. 1987). OPs are also used in production (Ramesh and Brown 1991) and scheduling (Ilhan et al. 2008) applications.

Many heuristics and exact algorithms for solving OPs have been proposed in the literature. A recent survey (Vansteenwegen et al. 2011) remarks that "the five-step heuristic of Chao et al. (1996a) clearly outperforms all above-mentioned [referring to those considered in the survey] heuristics." We use this heuristic as a point of departure in the development of a heuristic for the Smuggler Search Problem (SSP), where a simple node-deletion step is used to find an improving path. The Team OP (TOP), described in Chao et al. (1996b), is a generalization of the OP that allows for multiple routes through the transportation network (e.g., multiple homogeneous team members collecting rewards). As with the OP, many heuristics (Archetti et al. 2007, Chao et al. 1996a) and exact algorithms have been proposed for the TOP. Boussier et al. (2007) presents a branch-and-price algorithm that relies on a pricing step within the column generation phase, evaluating the reduced cost associated with routes not yet considered in the master problem. In order to do this efficiently, the reward for a path must be obtained cheaply (e.g., a table look-up). Other optimal solution procedures for the OP, TOP, and related problems use column generation approaches in a similar way (see, e.g., Butt and Ryan 1999). Branch-and-bound algorithms for the OP and related problems can also be found in the literature (see, e.g., Laporte and Martello 1990, Ramesh et al. 1992). Laporte and Martello (1990) present an enumerative algorithm where fathoming is accomplished by computing inexpensive upper bounds based on a binary knapsack problem. This is possible because the arc lengths and rewards are fixed values, conditions which do not necessarily hold in the GOP-RDR. Another approach (Ramesh et al. 1992) uses Lagrangian relaxation within a branch-and-bound procedure, where they relax the budget constraint and solve the resulting relaxation for fixed Lagrange multipliers using a polynomial time degree-constrained spanning tree algorithm; this is a technique that is not possible for the nonlinear GOP-RDR. Optimal

Lagrange multipliers are then computed via subgradient optimization.

Other generalizations of the OP have been considered in multi-objective problems where rewards can be functions of a number of attribute scores (see Silberholz and Golden 2010, Schilde et al. 2009, Wang et al. 2008, Wang et al. 1995), and where the arc length between nodes is determined by general cost functions (Ramesh and Brown 1991). The latter describes a generalized OP, where transit resources are not fixed values and the manner in which resources are consumed is determined by general resource expenditure functions. This generalized OP is solved with a four-phase heuristic which implements a series of node addition, node deletion, and node swapping steps, while using a bang-for-buck term that is based on a resource-to-reward ratio. No exact algorithms for this problem exist in the literature. The GOP-RDR further generalizes the OP by allowing node rewards to vary according a function of resources expended at the node.

More recently, Erdogan and Laporte (2013) studies another OP generalization where rewards can vary as a function of the number of visits to (or time spent at) a node. This problem is called the OP with Variable Profits (OPVP). This study considers two forms of the OPVP. In the first, a number of discrete reward collecting passes can be taken at each node so that more reward is collected as the number of passes increase. The second form considers node reward functions that increase with time spend at the node. The authors propose a solution approach where the nonlinear objective function is linearlized, similarly done in Royset and Sato (2010) in the context of optimal search, and the resulting mixed-integer program (MIP) is solved with a branch-and-cut algorithm. They report that linearizing the objective function is important so that they can introduce cuts in the MIP, a feature that is uncommon in nonlinear solvers. The OPVP is similar to the GOP-RDR in that rewards are allowed to vary as a function of resources (e.g., time) spent at each node. The GOP-RDR, however, is more general because arc lengths are allowed to vary as well. We will see in Chapter 3 that this important feature allows for modeling the situation where the nodes are in motion.

Some variants of the Vehicle Routing Problem (VRP) are related to the OP. A recent survey (Kumar and Panneerselvam 2012) and two books (Toth and Vigo 2002, Golden et al. 2008) contain detailed descriptions of most variants of the VRP as well as many popular heuristics and exact algorithms for solving them. A variant of the VRP which is closely related to the OP is the SVRP, which is sometimes called the VRP with Selective

Pickups (VRPSP). A relatively small proportion of the literature on the VRP considers the SVRP or other profit-maximizing VRPs where not all the customers have to be visited. In one such study, Privé et al. (2005) develops three heuristics to solve a VRP, where revenue generated from picking up recyclable containers enters in the objective function as an offset to travel cost. In another, Souffriau et al. (2013), contrasts the SVRP with Time Windows with the TOP.

The STSP and the MCP, which are similar to the OP, can also be found in the literature (see Laporte and Martello 1990, Gendreau et al. 1998, Butt and Cavalier 1994, Erkut and Zhang 1996). The GOP-RDR is closely related to the Red Cross blood-collection problem described by Yi (2003) where it is more beneficial to visit pickup locations later in the route because the visitation reward increases with time. This problem differs from the GOP-RDR in that, while rewards at each node depend on time (time being a resource consumed in transit between nodes), the activity of collecting the reward does not require resource consumption. Moreover, the arc lengths between nodes are fixed values in these problems. There appear to be no references in the literature which consider generalizations of the node rewards and arc lengths at the same time, nor does there appear to be problems where activity of collecting rewards at nodes is in direct competition with the activity of transiting between nodes. The GOP-RDR seems to be the first to consider these issues.

OPs and similar problems have been used to solve path-constrained optimal search problems. These problems are concerned with determining the best routing for search assets (searchers) in order to detect targets in some defined AOI. Path-constrained optimal search problems are known to be NP-hard (Trummel and Weisinger 1986). Benkoski et al. (1991) summarize much of the search theory literature through 1991. Recent research in optimal search has focused on discrete-time and -space models, developing various techniques to reduce solution times such as specialized branch-and-bound algorithms (see, e.g., Stewart 1979, Eagle and Yee 1990, Dell et al. 1996, Sato and Royset 2010), heuristics (see, e.g., Dell et al. 1996, Grundel 2005, Wong et al. 2005), and cutting-plane approaches (see, e.g., Royset and Sato 2010). Optimal search problem formulations have become more versatile in their ability to account for multiple cooperating searchers, multiple targets with different characteristics, as well as environmental effects on the search (see, e.g., Dell et al. 1996, Wong et al. 2005, Riehl et al. 2007, Royset and Sato 2010).

Coordinating search plans with interdiction assets is an important real-world consid-

eration. Few studies in the literature focus on this aspect of search planning. Kress et al. (2012) examines a discrete-time and -space stochastic Dynamic Programming (DP) approach to coordinate the efforts of a single arial search asset and a single surface interdiction asset. This model can, in principle, be solved by a backward DP algorithm, but becomes intractible when one considers problems arising from real-world scenarios. Accordingly, a greedy heuristic is proposed to obtain solutions. Related work has been done in studying search and action problems (see Sun 2009, Jin et al. 2006), however these models consider multi-role arial assets with some combination of sensing, intercept, and attack capabilities. Supporting the JIATFS mission requires the development of a new appraoch to modeling coordinated search and interdiction operations that lead to problems which are solvable for real-world scenarios.

Another important search planning consideration is planning in the presence of environmental uncertainty. Stone (1975, ch. 2) examines how a Search Allocation Problem (SAP) is affected by uncertain sensor performance. For this problem, uniformly optimal plans can be computed using the probability distributions of the uncertain sensor parameters. Search planning under uncertainty is considered in Evers et al. (2012), where a Robust OP (ROP) is proposed to find search plans that can withstand changes in the environment. While these studies consider search planning under uncertainty, their proposed techniques do not apply to the mixed-integer nonlinear SSP. We study how the SSP is affected by environmental uncertainty and we propose a multi-stage stochastic programming model that allows planners to develop contingency plans.

1.3 Contributions

This research presents the GOP-RDR, a generalization of the OP and several other related optimization problems, which appears to be the first OP to consider generalizations of the node rewards and arc lengths at the same time. We develop a specialized branch-and-bound (B&B) framework to compute solutions to the GOP-RDR.

Focusing specifically on the search application of the GOP-RDR, we develop the SSP, a novel path-constrained optimal search model in continuous time and space. The SSP requires fewer integer variables than classical discrete-time and -space optimal search models. We present a specialized B&B algorithm and three heuristics tailored to this model, which are able to quickly compute optimal search plans in scenarios that are on the scale of real-world counterdrug operations, a first in the field of optimal search.

We study five enhancements to the SSP, along with tailored solution procedures, that can be used together or separately to model various real-world search planning scenarios. With these enhancements planners can account for complex target motion, multi-vehicle search planning, high uncertainty in target motion, the coordination of search and interdiction efforts, and multi-period search planning over time. The SSP, combined with these enhancements, is the first optimal search model to consider all of these issues.

We study how environmental conditions can impact optimal solutions to the SSP, and highlight conditions where plans are most affected. Since environmental conditions are uncertain by nature, we investigate how evolving uncertainty in planning data can be addressed by a multi-stage stochastic programming model. This model can yield improved plans by incorporating environmental uncertainty. This appears to be the first multi-stage optimal search model that is able to handle problems on the scale of those encountered in real-world counterdrug operaions.

1.4 Disclaimer

While this research aims to support JIATFS in conducting search and interdiction planning for counterdrug operations, information presented and views expressed in this dissertation do not reflect the official policy, position, or practices of JIATFS or any other U.S. Government organization. Mission details and data used throughout are obtained from unclassified and unrestricted sources. Mission details are derived from United States Southern Command (2013), Joint Interagency Task Force South (2013), Munsing and Lamb (2011), and the author's interpretations thereof. Data are derived from Munsing and Lamb (2011), United States Coast Guard (2013), GlobalSecurity.org (2013), and approximations based on the author's personal experience.

1.5 Organization

This dissertation is organized as follows. We formulate the GOP-RDR and provide a branch-and-bound framework for obtaining solutions in the next chapter. In Chapter 3, we formulate the SSP, describe two heuristics that provide initial solutions to the branch-and-bound algorithm, and present numerical results, comparing branch-and-bound solutions to solutions obtained by mixed-integer nonlinear programming solvers. Chapter 4 describes SSP enhancements that allow practitioners to model realistic search planning problems and demonstrates how they can be applied using a baseline scenario. In Chapter 5 we

investigate the environmental sensitivity of optimal search plans and demonstrate how to account for changes in the environment using multi-stage stochastic programming. We conclude with final remarks in Chapter 6.

CHAPTER 2:

Generalized Orienteering Problem with Resource Dependent Rewards

2.1 Formulation

Before formulating the Generalized Orienteering Problem with Resource Dependent Rewards (GOP-RDR) we begin with a standard Orienteering Problem (OP) formulation (Vansteenwegen et al. 2011), which we will use as a stepping stone for generalization. In an OP, a vehicle is routed through a transportation network, collecting rewards at each node. Let $G = (N, A)$ be the directed graph that models this transportation network, where $N = \{0, 1, \ldots, n+1\}$ is the node set and A is the arc set. Nodes 0 and $n+1$ are the vehicle's home station and recovery location, respectively; not necessarily the same physical location. We assume that all arcs incident to node 0 are outbound arcs, and that all arcs incident to node $n+1$ are inbound arcs. For notational convenience, we define $\hat{N} = N \backslash \{0, n+1\}$ as the set of nodes excluding the home station, node 0, and the recovery location, node $n+1$. A reward q_i is collected at each node $i \in \hat{N}$. Traversing any arc $(i, j) \in A$ consumes a fixed resource $\bar{t}_{i,j}$. Total resource expenditure is limited by T. We model the vehicle path on G using the binary variables $x_{i,j}$, where $x_{i,j}$ takes on value 1 when arc (i, j) is in the path, and 0 otherwise. We then obtain the following OP formulation.

Problem O:

$$\max_{\boldsymbol{x}} \quad \sum_{j \in \hat{N}} q_j \left(\sum_{i:(i,j) \in A} x_{i,j} \right) \tag{2.1a}$$

$$\text{s.t.} \quad \sum_{(i,j) \in A} \bar{t}_{i,j} x_{i,j} \leq T \tag{2.1b}$$

$$\sum_{i:(i,j) \in A} x_{i,j} - \sum_{i:(j,i) \in A} x_{j,i} = \begin{cases} -1, & j = 0 \\ 0, & \forall j \in \hat{N} \\ 1, & j = n+1 \end{cases} \tag{2.1c}$$

$$\sum_{i:(i,j)\in A} x_{i,j} \leq 1, \qquad \forall j \in N \tag{2.1d}$$

$$\sum_{\substack{(i,j)\in A:\\ i,j\in N'}} x_{i,j} \leq |N'| - 1, \quad \forall N' \subseteq \hat{N}, N' \neq \emptyset \tag{2.1e}$$

$$x_{i,j} \in \{0,1\}, \qquad \forall (i,j) \in A \tag{2.1f}$$

The objective (2.1a) accumulates rewards along the path. Constraint (2.1b) ensures the resources $\bar{t}_{i,j}$ expended along the path do not exceed the resource limit T. Constraints (2.1c) maintain a balanced network flow that starts at the home station and ends at the recovery location. Constraints (2.1d) ensure nodes are visited at most once. Constraints (2.1e) are the subtour elimination constraints proposed in Desrochers and Laporte (1991), which are known to yield relatively tight linear programming relaxations (see Toth and Vigo 2002, ch. 1). Constraints (2.1f) require that variables $x_{i,j}$ be binary.

For notational convenience we use the auxiliary binary variable y_j, which is uniquely determined by variables $x_{i,j}$. Variable y_j takes on value 1 when node j is in the path, and 0 otherwise; i.e.,

$$y_0 = 1, \qquad y_j = \sum_{i:(i,j)\in A} x_{i,j}, \forall j \in N\backslash\{0\} \tag{2.2}$$

We denote by $\boldsymbol{x}$ the vector of *path* variables $\{x_{i,j} : (i,j) \in A\}$. We denote by $\boldsymbol{y}$ the vector of *node visitation* variables $\{y_j : j \in N\}$ and represent (2.2) with the expression $\boldsymbol{y} = \boldsymbol{\Gamma x}$ for an appropriately selected matrix $\boldsymbol{\Gamma}$. We define $\mathbb{X}$ as the set of paths that satisfy (2.1c), (2.1d), (2.1e), and (2.1f).

We generalize $\mathbf{O}$ to construct the GOP-RDR as follows. A visit to any node $i \in \{1,\ldots,n\}$ is rewarded at the expense of consuming r *dwell* resources $\boldsymbol{d}_i \in \mathbb{R}^r$. Similarly, *transit* resources $\boldsymbol{t}_{i,j} \in \mathbb{R}^r$ are consumed when traveling directly from node i to node j. Resources may represent, for example, various consumables such as time, fuel, and/or money. We also include *auxiliary* resource variables $\boldsymbol{a}_i \in \mathbb{R}^r$, which, when consumed at nodes, yield no reward. These variables may be used, for example, to track the accumulation of dwell and transit resources expended along the path. Let $\boldsymbol{a}, \boldsymbol{d} \in \mathbb{R}^{r(n+2)}$, and $\boldsymbol{t} \in \mathbb{R}^{r|A|}$ denote vectors of resource variables; i.e., $\boldsymbol{a} = (\boldsymbol{a}_0^T, \boldsymbol{a}_1^T, \ldots, \boldsymbol{a}_{n+1}^T)^T$ and similarly for $\boldsymbol{d}$ and $\boldsymbol{t}$. A vehicle may collect rewards according to the concave utility function $f(\boldsymbol{d}) : \mathbb{R}^{r(n+2)} \mapsto \mathbb{R}$, with $f(\mathbf{0}) = 0$. We assume without loss of generality that no reward is possible at nodes

0 and $n+1$. The vehicle path through G must obey η resource expenditure laws on each arc $(i,j) \in A$ denoted by the functions $h_{i,j}(\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{d}_i, \boldsymbol{d}_j, \boldsymbol{t}_{i,j}) : \mathbb{R}^{5r} \mapsto \mathbb{R}^{\eta}$. The resource expenditure laws account for applications where arc lengths are allowed to vary. In the static network considered in **O**, $h_{i,j} = ||\bar{t}_{i,j} - t_{i,j}||$ for fixed arc lengths $\bar{t}_{i,j}$, but in Chapter 3 arc lengths are not fixed because the nodes of the network are in motion. The vehicle path must also obey resource expenditure laws at each node $i \in N$ denoted by the functions $g_i(\boldsymbol{a}_i, \boldsymbol{d}_i) : \mathbb{R}^{2r} \mapsto \mathbb{R}^{\gamma}$ and $m_i(\boldsymbol{a}_i, \boldsymbol{d}_i) : \mathbb{R}^{2r} \mapsto \mathbb{R}^{\mu}$. We assume that functions $h_{i,j}$ and g_j are convex, and functions m_j are affine. The vehicle path must be such that total resource expenditure stays within the resource limits defined by $\boldsymbol{T} \in \mathbb{R}^r$. Let the matrix $\boldsymbol{Y} \in \mathbb{R}^{r(n+2) \times r(n+2)}$ be the diagonal matrix $\text{diag}(y_0\mathbf{0}, y_1\mathbf{1}, y_2\mathbf{1}, \ldots, y_n\mathbf{1}, y_{n+1}\mathbf{0})$, where $\mathbf{0}, \mathbf{1} \in \mathbb{R}^r$ are vectors of 0s and 1s respectively. We note that the expression $\boldsymbol{Y}\boldsymbol{d}$ simply returns the dwell resource vector associated with reward collection nodes in the path $\boldsymbol{x}$. We now state the GOP-RDR.

Problem P:

Sets

N	nodes: $i, j \in \{0, 1, \ldots, n+1\}$
A	arcs
$\mathbb{X}$	paths that satisfy (2.1c), (2.1d), (2.1e), and (2.1f)

Parameters

$\boldsymbol{T} \in \mathbb{R}^r$	resource expenditure limits
$\boldsymbol{\Gamma} \in \mathbb{R}^{(n+2) \times \lvert A \rvert}$	path-to-node visitation mapping matrix representing (2.2)

Functions

$f : \mathbb{R}^{r(n+2)} \mapsto \mathbb{R}$	concave reward collection objective function
$h_{i,j} : \mathbb{R}^{5r} \mapsto \mathbb{R}^{\eta}$	convex resource expenditure law functions
$g_{i,j} : \mathbb{R}^{2r} \mapsto \mathbb{R}^{\gamma}$	convex node resource expenditure law functions
$m_{i,j} : \mathbb{R}^{2r} \mapsto \mathbb{R}^{\mu}$	affine node resource expenditure law functions

Variables

$\boldsymbol{a}_i$	node i auxiliary resource variable
$\boldsymbol{d}_i$	node i dwell resource variable
$\boldsymbol{t}_{i,j}$	arc (i,j) transit resource variable
$x_{i,j}$	arc (i,j) binary path variable
y_i	node i binary visitation variable;
	$\boldsymbol{Y}$ is a diagonal matrix of these variables

Formulation

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t},\boldsymbol{x},\boldsymbol{y}} \quad f(\boldsymbol{Y}\boldsymbol{d}) \tag{2.3a}$$

$$\text{s.t.} \quad h_{i,j}(\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{d}_i, \boldsymbol{d}_j, \boldsymbol{t}_{i,j}) x_{i,j} \leq 0, \quad \forall (i,j) \in A \tag{2.3b}$$

$$\sum_{j \in N} \boldsymbol{d}_j + \sum_{(i,j) \in A} \boldsymbol{t}_{i,j} \leq \boldsymbol{T} \tag{2.3c}$$

$$g_j(\boldsymbol{a}_j, \boldsymbol{d}_j) \leq 0, \quad \forall j \in N \tag{2.3d}$$

$$m_j(\boldsymbol{a}_j, \boldsymbol{d}_j) = 0, \quad \forall j \in N \tag{2.3e}$$

$$\boldsymbol{a}_j, \boldsymbol{d}_j \geq 0, \quad \forall j \in N \tag{2.3f}$$

$$\boldsymbol{t}_{i,j} \geq 0, \quad \forall (i,j) \in A \tag{2.3g}$$

$$\boldsymbol{y} = \boldsymbol{\Gamma}\boldsymbol{x} \tag{2.3h}$$

$$\boldsymbol{x} \in \mathbb{X} \tag{2.3i}$$

The objective (2.3a) maximizes the reward collected along the path. Constraints (2.3b) enforce resource expenditure laws on each arc. Constraint (2.3c) ensures that total resource expenditure is within the prescribed limits. Constraints (2.3d) and (2.3e) enforce resource expenditure laws at each node. Concavity of f makes it desirable to consume resources $\boldsymbol{d}$. Constraints (2.3c) make it undesirable to consume resources $\boldsymbol{t}$. However, $\boldsymbol{d}$ and $\boldsymbol{t}$, along with $\boldsymbol{a}$, are related through constraints (2.3b) so it may not be possible to consume $\boldsymbol{t} = \boldsymbol{0}$ transit resources. We observe that when node j is not in the path $\boldsymbol{x}$, (2.3d) and (2.3e) are vacuous because $\boldsymbol{a}_j$ and $\boldsymbol{d}_j$ can be chosen arbitrarily to satisfy these constraints provided total resource expenditure (2.3c) is not exceeded. Constraint (2.3c)

makes $\boldsymbol{d}_j = \mathbf{0}$ desirable in this situation because a higher reward is obtained by consuming more dwell resources at visited nodes. We assume that $\boldsymbol{d}_j = \mathbf{0}$ is always feasible. Similarly, when arc (i, j) is not in the path $\boldsymbol{x}$, constraint (2.3b) is inactive and the resource constraint (2.3c) forces $\boldsymbol{t}_{i,j} = \mathbf{0}$. Constraints (2.3f) and (2.3g) require nonnegative resource expenditure. Binary visitation and path variables are set by (2.3h) and (2.3i) respectively.

Proposition 1. *The GOP-RDR is NP-hard.*

Proof. Since the OP is known to be NP-hard (Golden et al. 1987, Laporte and Martello 1990) it suffices to show that the GOP-RDR contains an OP as a special case. We begin with an arbitrary instance of **O** and introduce continuous variables $\boldsymbol{a}, \boldsymbol{d}$ and $\boldsymbol{t}$, and binary vector $\boldsymbol{y}$ as in (2.2). We define functions $h_{i,j}$, g_j, and m_j as follows.

$$h_{i,j}(\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{d}_i, \boldsymbol{d}_j, \boldsymbol{t}_{i,j}) = ||\bar{t}_{i,j} - t_{i,j}||, \quad \forall (i,j) \in A \tag{2.4}$$

$$g_j(\boldsymbol{a}_j, \boldsymbol{d}_j) = m_j(\boldsymbol{a}_j, \boldsymbol{d}_j) = ||d_j - 1||, \quad \forall j \in N \tag{2.5}$$

We add to **O** the constraints (2.3b), (2.3d), and (2.3e). Due to (2.5), $\boldsymbol{a}$ is free to take on any nonnegative value and $d_j = 1, \forall j \in N$. Using (2.2), the objective function (2.1a) of **O** is equivalently stated as follows.

$$\sum_{j \in \hat{N}} q_j \left(\sum_{i:(i,j) \in A} x_{i,j} \right) = \sum_{j \in \hat{N}} q_j y_j = \sum_{j \in \hat{N}} q_j d_j y_j = f(\boldsymbol{Y}\boldsymbol{d})$$

Setting $\bar{T} = \boldsymbol{T} - |N|$ and observing that due to (2.3b) and (2.4),

$$t_{i,j} = \begin{cases} \bar{t}_{i,j}, & x_{i,j} = 1 \\ 0, & x_{i,j} = 0 \end{cases}$$

and constraints (2.1b) are equivalent to (2.3c). We immediately see that since the only remaining free variables in the resulting GOP-RDR are $\boldsymbol{x}$, the optimal $\boldsymbol{x}$ in this GOP-RDR are also optimal in **O**. Moreover, since the objective functions are equal for any common $\boldsymbol{x}$, this GOP-RDR solves **O**. □

The MINLP **P** has a non-convex continuous relaxation. When this is the case most

MINLP solvers such as DICOPT (Grossmann et al. 2013) and BONMIN (Vigerske 2013) provide no guarantees of finding globally optimal solutions. The problem can be convexified with a Big-M reformulation. For example, (2.3b) can be reformulated as

$$h_{i,j}(\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{d}_i, \boldsymbol{d}_j, \boldsymbol{t}_{i,j}) \leq M(1 - x_{i,j}), \forall (i,j) \in A,$$

for M sufficiently large. We discuss Big-M reformulation and show numerical results in the context of the SSP in Chapter 3. Another approach is to use a B&B-based MINLP solver such as BARON that uses convexifying techniques at each node of a B&B enumeration tree to obtain globally optimal solutions (Sahinidis and Tawarmalani 2013, Tawarmalani and Sahinidis 2004). This approach is not pursued here, rather we use the underlying structure of $\mathbf{P}$ as a basis for computing solutions.

With these matters in mind, we now proceed to describe a B&B approach that utilizes convex relaxation problems, avoids Big-M reformulations, and capitalizes on the underlying structure of $\mathbf{P}$ as a basis for branching and pruning. We introduce the notation $G(\boldsymbol{x}) = (N(\boldsymbol{x}), A(\boldsymbol{x}))$, where $N(\boldsymbol{x}) = \{j \in N : y_j = 1; y_0, = 1, \boldsymbol{y} = \boldsymbol{\Gamma x}\}$ and $A(\boldsymbol{x}) = \{(i,j) \in A : x_{i,j} = 1\}$. For any path $\boldsymbol{x} \in \mathbb{X}$, $\mathbf{P}$ can be expressed as the following convex, fixed-path NLP.

Problem $\mathbf{P}(\boldsymbol{x})$:

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t}} \quad f(\boldsymbol{Y d}) \tag{2.6a}$$

$$\text{s.t.} \quad h_{i,j}(\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{d}_i, \boldsymbol{d}_j, \boldsymbol{t}_{i,j}) \leq 0, \quad \forall (i,j) \in A(\boldsymbol{x}) \tag{2.6b}$$

$$\sum_{j \in N(\boldsymbol{x})} \boldsymbol{d}_j + \sum_{(i,j) \in A(\boldsymbol{x})} \boldsymbol{t}_{i,j} \leq \boldsymbol{T} \tag{2.6c}$$

$$g_j(\boldsymbol{a}_j, \boldsymbol{d}_j) \leq 0, \quad \forall j \in N(\boldsymbol{x}) \tag{2.6d}$$

$$m_j(\boldsymbol{a}_j, \boldsymbol{d}_j) = 0, \quad \forall j \in N(\boldsymbol{x}) \tag{2.6e}$$

$$\boldsymbol{a}_j, \boldsymbol{d}_j \geq 0, \quad \forall j \in N(\boldsymbol{x}) \tag{2.6f}$$

$$\boldsymbol{t}_{i,j} \geq 0, \quad \forall (i,j) \in A(\boldsymbol{x}) \tag{2.6g}$$

When $\boldsymbol{x}$ is fixed, $\boldsymbol{y}$ can be computed by (2.2). Variables $\boldsymbol{a}_j$ and $\boldsymbol{d}_j$ corresponding to unvisited nodes are removed from the problem. Similarly, variables $\boldsymbol{t}_{i,j}$ corresponding to

arcs not traversed are eliminated. The resulting convex NLP in the remaining variables $\boldsymbol{a}, \boldsymbol{d}$, and $\boldsymbol{t}$ is efficiently solved by standard NLP solvers such as MINOS (Murtagh et al. 2013) and SNOPT (Gill et al. 2013). If we are able to enumerate all possible paths $\boldsymbol{x} \in \mathbb{X}$ and solve $\mathbf{P}(\boldsymbol{x})$ for each, we are assured to find an optimal solution to $\mathbf{P}$.

2.2 Branch-and-Bound Framework

Observing that $\boldsymbol{d}$ contributes to reward collection and $\boldsymbol{t}$ consumes resources without reward, clearly nonzero values of $\boldsymbol{d}$ and small values of $\boldsymbol{t}$ are always desired. We construct the matrix $\tilde{\boldsymbol{I}}$ by taking an $r(n+2) \times r(n+2)$ identity matrix, and setting the first r diagonal entries and the last r diagonal entries to zero. The expression $\tilde{\boldsymbol{I}}\boldsymbol{d}$ returns the vector of dwell resources, setting home and recovery dwell resources to zero. We define $\boldsymbol{\delta}_{i,j} \in \mathbb{R}^r$ as the smallest possible resource expenditure between node i and node j. We assume that $\boldsymbol{T} \geq \boldsymbol{\delta}_{0,j} + \boldsymbol{\delta}_{j,n+1}, \forall j \in N$. Any node where this assumption does not hold is removed from the network because it cannot be on any feasible path $\boldsymbol{x}$ between node 0 and node $n+1$. If we consider $\mathbf{P}(\boldsymbol{x})$ and allow reward to be collected at every node with no transit resource expenditure on any arc, we obtain the following relaxed NLP.

Problem RP(0):

$$\max_{\boldsymbol{a},\boldsymbol{d}} \quad f(\tilde{\boldsymbol{I}}\boldsymbol{d}) \tag{2.7a}$$

$$\text{s.t.} \quad \sum_{j \in N} \boldsymbol{d}_j \leq \boldsymbol{T} - \min_{\substack{j \in N: \\ (0,j) \in A}} \{\boldsymbol{\delta}_{0,j}\} - \min_{\substack{j \in N: \\ (j,n+1) \in A}} \{\boldsymbol{\delta}_{j,n+1}\} \tag{2.7b}$$

$$\boldsymbol{d}_j \leq \boldsymbol{T} - \boldsymbol{\delta}_{0,j} - \boldsymbol{\delta}_{j,n+1}, \qquad \forall j \in \hat{N} \tag{2.7c}$$

(2.3d), (2.3e), and (2.3f)

We observe that the resource limit decrement on the right hand side of (2.7b),

$$\min_{\substack{j \in N: \\ (0,j) \in A}} \{\boldsymbol{\delta}_{0,j}\} + \min_{\substack{j \in N: \\ (j,n+1) \in A}} \{\boldsymbol{\delta}_{j,n+1}\}, \tag{2.8}$$

is a lower bound on $\boldsymbol{t}_{0,j} + \boldsymbol{t}_{j,n+1}, \forall j \in \hat{N}$. A path that visits any nonempty subset of nodes $N' \subseteq \hat{N} : N' \neq \emptyset$ consumes at least (2.8) transit resources. Constraint (2.7c) simply requires that the upper bound on dwell resources at each node $\boldsymbol{d}_j$ is decremented by the

minimum resource consumption on the path $0 \to j \to n+1$.

Proposition 2. **RP(0)** *is a relaxation of* **P**.

Proof. **P** is obtained by adding constraints (2.3b), (2.3g), (2.3h), and (2.3i) to **RP(0)**, while restricting reward collection (2.3a) to nodes in the path, and incurring a transit resource expenditure in (2.3c) that is no less than (2.8). □

We denote the optimal objective function values of **P**, $\mathbf{P}(\boldsymbol{x})$, and **RP(0)** by Z^*, $Z(\boldsymbol{x})^*$, and $Z(\mathbf{0})^*$, respectively, and state the following result.

Proposition 3. $Z(\mathbf{0})^* \geq Z^* \geq Z(\boldsymbol{x})^*, \forall \boldsymbol{x} \in \mathbb{X}$.

Proof. The result follows from the fact that **RP(0)** is a relaxation of **P** and that $Z^* = \max_{\boldsymbol{x} \in \mathbb{X}} Z(\boldsymbol{x})^*$. □

In order to obtain useful bounds on **P**, we introduce the notion of a partial path. We define a partial path $\hat{\boldsymbol{x}}_\ell$ to be the binary vector satisfying constraints (2.1d), (2.1e), and (2.1f), while constraints (2.1c) are satisfied for all nodes except the recovery location $n+1$ and the last node ℓ visited. We denote by $N^\times(\hat{\boldsymbol{x}}_\ell)$ the set of nodes that, if added to the current partial path $\hat{\boldsymbol{x}}_\ell$, would violate one or more constraints in **P**. Any node $j \in N^\times(\hat{\boldsymbol{x}}_\ell)$ cannot be considered in later extensions to any extension of $\hat{\boldsymbol{x}}_\ell$. Therefore, $N^\times(\hat{\boldsymbol{x}}_\ell)$ does not reduce in size as the partial path $\hat{\boldsymbol{x}}_\ell$ is extended. Formally, $N^\times(\hat{\boldsymbol{x}}_\ell) \subseteq N^\times(\hat{\boldsymbol{x}}_k), \forall k : (\ell, k) \in A$. In general $N^\times(\hat{\boldsymbol{x}}_\ell) = \emptyset$, but, depending on the structure of h, g and/or m, this set may be nonempty. In Chapter 3, this is discussed further in the context of the SSP and rules for constructing $N^\times(\hat{\boldsymbol{x}}_\ell)$ are presented. We define the indicator parameter I_ℓ that takes on value 1 when $\ell = n+1$, and 0 otherwise. We denote by $\tilde{\boldsymbol{T}}_j(\hat{\boldsymbol{x}}_\ell)$ the optimistic dwell resource expenditure for search region j that is not in the partial path $\hat{\boldsymbol{x}}_\ell$.

$$\tilde{\boldsymbol{T}}_j(\hat{\boldsymbol{x}}_\ell) = \max \left\{ \mathbf{0}, \boldsymbol{T} - \boldsymbol{\delta}_{\ell,j} - \boldsymbol{\delta}_{j,n+1} - \sum_{(i,i') \in A(\hat{\boldsymbol{x}}_\ell)} \boldsymbol{\delta}_{i,i'} \right\} \tag{2.9}$$

For any partial path $\hat{\boldsymbol{x}}_\ell$ we have the following convex partial path relaxation NLP.

Problem RP($\hat{\boldsymbol{x}}_\ell$):

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t}} \quad f(\tilde{\boldsymbol{I}}\boldsymbol{d}) \tag{2.10a}$$

$$\text{s.t.} \quad h_{i,j}(\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{d}_i, \boldsymbol{d}_j, \boldsymbol{t}_{i,j}) \leq 0, \qquad \forall (i,j) \in A(\hat{\boldsymbol{x}}_\ell) \tag{2.10b}$$

$$\sum_{j \in N} \boldsymbol{d}_j + \sum_{(i,j) \in A(\hat{\boldsymbol{x}}_\ell)} \boldsymbol{t}_{i,j} \leq \boldsymbol{T} - (1 - I_\ell)\boldsymbol{\delta}_{\ell,n+1}, \tag{2.10c}$$

$$\boldsymbol{d}_j \leq \tilde{\boldsymbol{T}}_j(\hat{\boldsymbol{x}}_\ell), \qquad \forall j \in N \backslash N(\hat{\boldsymbol{x}}_\ell) \tag{2.10d}$$

$$I_\ell \boldsymbol{d}_j = 0, \qquad \forall j \in N \backslash N(\hat{\boldsymbol{x}}_\ell) \tag{2.10e}$$

$$\boldsymbol{d}_j = 0, \qquad \forall j \in N^{\times}(\hat{\boldsymbol{x}}_\ell) \tag{2.10f}$$

$$\boldsymbol{t}_{i,j} \geq 0, \qquad \forall (i,j) \in A(\hat{\boldsymbol{x}}_\ell) \tag{2.10g}$$

(2.3d), (2.3e), and (2.3f)

We note that when $I_\ell = 1$, the path is complete, and $\mathbf{P}(\boldsymbol{x})$ and $\mathbf{RP}(\hat{\boldsymbol{x}}_\ell)$ are equivalent. Conversely, when $I_\ell = 0$, the path is a partial path, (2.10e) is inactive and resources associated with unvisited nodes are allowed to take on nonzero values, the right hand side of (2.10c) is decremented by $\boldsymbol{\delta}_{\ell,n+1}$, and the right hand side of (2.10d) bounds dwell time associated with unvisited nodes based on minimum travel time resource expenditure. We observe that constraints (2.10f) only rule out dwell time associated with infeasible partial path extensions.

Suppose that the partial path $\hat{\boldsymbol{x}}_\ell$ is extended by adding any arc $(\ell, k) \in A$ to the path as shown in Figure 2.1. In this depiction, the minimum resource expenditure $\boldsymbol{\delta}_{\ell,n+1}$ in the partial path $\hat{\boldsymbol{x}}_\ell$ is no larger than the transit resource expenditure $\boldsymbol{t}_{\ell,k} + \boldsymbol{\delta}_{\ell,n+1}$ in the partial path $\hat{\boldsymbol{x}}_k$. Next we show results that support building successive restrictions of $\mathbf{RP}(\hat{\boldsymbol{x}}_\ell)$ by adding to the partial path.

Theorem 1. **RP($\hat{\boldsymbol{x}}_k$)** *is a restriction of* **RP($\hat{\boldsymbol{x}}_\ell$)**, $\forall k : (\ell, k) \in A$.

Proof. We observe that if $\ell = n + 1$, then $\{k : (\ell, k) \in A\} = \emptyset$. Suppose $\ell \in N \backslash \{n + 1\}$. Adding node k and arc (ℓ, k) to the partial path, we add a block of constraints to (2.10b) and variable $\boldsymbol{t}_{\ell,k}$ in (2.10c) and (2.10g). Since $N^{\times}(\hat{\boldsymbol{x}}_\ell) \subseteq N^{\times}(\hat{\boldsymbol{x}}_k)$, no constraints (2.10f) are removed. Lastly, since the increase in resource expenditure along the new partial path is at least $\boldsymbol{\delta}_{\ell,n+1}$, the result follows. □

Let $Z(\hat{\boldsymbol{x}}_\ell)^*$ be the optimal objective function value of $\mathbf{RP}(\hat{\boldsymbol{x}}_\ell)$.

Corollary 1. *$Z(\hat{\boldsymbol{x}}_\ell)^* \geq Z(\hat{\boldsymbol{x}}_k)^*, \forall k : (\ell, k) \in A$.*

Proof. If $\ell = n+1$, then $\{k : (\ell, k) \in A\} = \emptyset$. Alternatively, if $\ell \in N \backslash \{n+1\}$ the result follows by Theorem 1. □

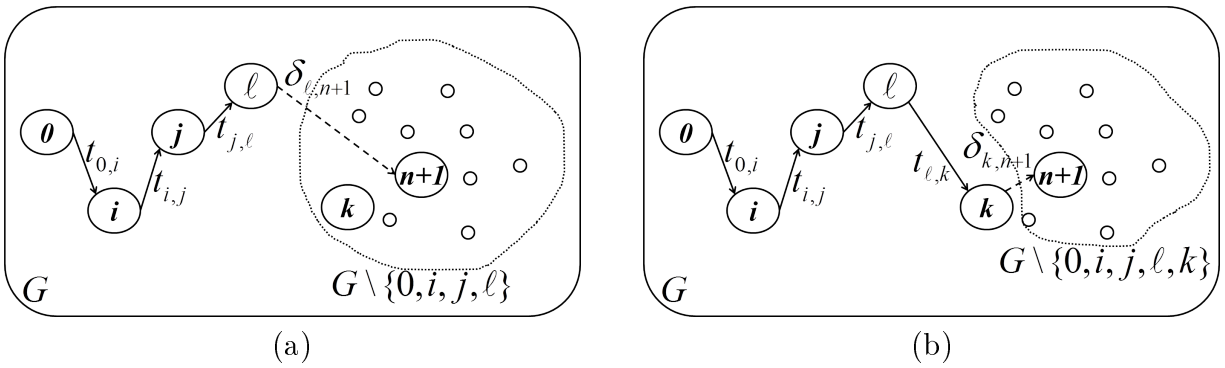


Figure 2.1: **Partial path resource expenditure** - (a) Partial path $\hat{\boldsymbol{x}}_\ell$. (b) Partial path $\hat{\boldsymbol{x}}_k$. Transit resource expenditure along partial path $\hat{\boldsymbol{x}}_\ell$ is no larger than transit resource expenditure along path $\hat{\boldsymbol{x}}_k$.

We present a GOP-RDR B&B framework that begins at the home station and forms partial paths by adding nodes to a path sequentially, solving restrictions of $\mathbf{RP}(\hat{\boldsymbol{x}}_\ell)$ along the way. We denote by $\hat{\boldsymbol{X}}$ the set of all possible partial paths $\hat{\boldsymbol{x}}$. Let $l \in \{0, 1, \ldots, n+1\}$ denote the level of the B&B enumeration tree, and let $L_l \subseteq N$ be the set of nodes yet to be considered at level l. We define the set $\Omega \subseteq \hat{\boldsymbol{X}} \times N$ to be a subset of partial path and B&B enumeration tree level pairs. Let $\epsilon \geq 0$ be the absolute optimality gap stopping tolerance.

Algorithm B&B:

1. ***Initialization***
   Initialize $\ell = 0$; $\boldsymbol{x}^* = \hat{\boldsymbol{x}}_\ell = \mathbf{0}$; lower bound $LB = 0$; $l = 0$; $L_0 = \emptyset$; $L_k = N, \forall k = 1, \ldots, n+1$; and $\Omega = \{(\mathbf{0}, 0)\}$. Solve $\mathbf{RP}(\mathbf{0})$. If $\mathbf{RP}(\mathbf{0})$ is infeasible, then stop; $\mathbf{P}$ is infeasible. Otherwise, initialize upper bound $UB = Z(\mathbf{0})^*$.
2. ***Branching***
   If $UB - LB \leq \epsilon$, then stop and return $\boldsymbol{x}^*$. Otherwise, choose $(\hat{\boldsymbol{x}}_\ell, l) \in \Omega$. Add node

j to partial path $\hat{\boldsymbol{x}}_\ell$ to form the extended partial path $\hat{\boldsymbol{x}}_j \in \hat{\boldsymbol{X}}$ that contains arc (ℓ, j). Add $\{(\hat{\boldsymbol{x}}_j, l+1)\}$ to Ω. Remove j from L_{l+1}. Solve $\mathbf{RP}(\hat{\boldsymbol{x}}_j)$.

3. ***Lower-Bounding***
   If $j = n+1$ and $LB < Z(\hat{\boldsymbol{x}}_j)^*$ and $\mathbf{RP}(\hat{\boldsymbol{x}}_j)$ is feasible, then set $LB = Z(\hat{\boldsymbol{x}}_j)^*$ and $\boldsymbol{x}^* = \hat{\boldsymbol{x}}_j$ as the best complete path found thus far.
4. ***Upper-Bounding***
   If $L_{l+1} = \emptyset$ then set $UB = \max\{Z^*_{n+1}, Z^*_{l+1}\}$, where Z^*_{n+1} is the largest value $Z(\boldsymbol{x})^*$ of all complete paths explored thusfar and Z^*_{l+1} is the largest value $Z(\hat{\boldsymbol{x}}_j)^*$ on level $l+1$ of the B&B enumeration tree.
5. ***Fathoming***
   If $Z(\hat{\boldsymbol{x}}_j)^* < LB$ or if $\mathbf{RP}(\hat{\boldsymbol{x}}_j)$ is infeasible, then fathom partial path $\hat{\boldsymbol{x}}_j$ by removing from Ω all elements $(\hat{\boldsymbol{x}}_{\tilde{j}}, \tilde{l})$, where $\hat{\boldsymbol{x}}_j$ is a subpath of $\hat{\boldsymbol{x}}_{\tilde{j}}$ and $\tilde{l} > l$.
6. ***Iteration***
   Return to step 2.

This algorithm is guaranteed to converge after solving a finite number of partial path relaxation problems $\mathbf{RP}(\hat{\boldsymbol{x}}_\ell)$. If $\epsilon = 0$ and step 5 is eliminated, the algorithm would simply enumerate partial paths, compute the solution to $\mathbf{RP}(\hat{\boldsymbol{x}}_\ell)$ for each, and return the optimal solution after considering all partial paths. Since fathoming, in step 5, only eliminates suboptimal and infeasible paths, the algorithm is guaranteed to produce a path with an objective function value that is within ϵ of the optimal objective function value.

The algorithm can be accelerated by obtaining an initial feasible solution $\boldsymbol{x}$ that produces a better lower bound in step 1, thereby allowing fathoming in step 5 to occur more rapidly. To this end, we provide a specialized heuristic for the SSP in Section 3.2.1. We do not prescribe the nature of branching to be performed in step 2. Numerical results discussed in Section 3.3 use depth-first-search, but other branching strategies can also be used (see, for example, Ramesh et al. 1992, Toth and Vigo 2002).

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Smuggler Search Problem

The Smuggler Search Problem (SSP) is a special case of a GOP-RDR that arises in challenging real-world search operations. The SSP serves as a model to support JIATFS in detecting and interdicting the flow of illicit traffickers in international waters. To accomplish this mission, planners must employ a limited number of search assets as effectively as possible, under strict resource constraints, as they respond to uncertain estimates of how illicit traffickers move in the AOI.

Many approaches to solving path-constrained optimal search problems using a discrete-time and -space model can be found in the literature (Dell et al. 1996, Eagle and Yee 1990, Grundel 2005, Royset and Sato 2010, Sato and Royset 2010). These approaches yield large models when considering the size of the JIATFS AOI. For example, given that the JIATFS AOI covers 42 million square miles (Munsing and Lamb 2011) and conservatively estimating that target activity occurs in only 1% of the AOI, even a coarse discretization of 10,000-nm^2-area cells and hour-long time periods over a 24-hour mission day yields a network with over 1,000 nodes. The state-of-the-art OP algorithms are reported to solve problems with fewer than 500 nodes (Vansteenwegen et al. 2011). Similarly, discrete-time and -space optimal search models of this size cannot be solved to optimality in a reasonable amount of time using current methods (Royset and Sato 2010). We proceed to formulate the SSP, a novel path-constrained optimal search model in continuous time and space, which avoids large models caused by discretization.

Consider a planning scenario where a searcher is to be routed throughout an AOI to detect multiple moving targets. The search controller has, based on planning factors, the information listed in Table 3.1.

Maximum cruise speed of the searcher while in transit	V
Speed of the searcher while performing search actions	$\hat{V}$
Searcher sensor sweep width	W
Searcher endurance time limit	T
Scenario time limit	D

Table 3.1: **Search planning factors**

Suppose that there is uncertainty with regard to where and when each target departs,

where the target arrives, and the value of detecting the target, but that the nature of the intelligence allows the search controller to estimate these values within some range of uncertainty. These data are listed in Table 3.2. We refer to data listed in Tables 3.1 and 3.2 collectively as *planning data*. Figure 3.1 illustrates the assessed target data for a two-target scenario on a map.

Number of targets	n
Speed of target j	U_j
Expected departure time of target j	τ_j
Time uncertainty range of target j	$\tilde{\tau}_j$
Expected departure location of target j	$\boldsymbol{\rho}_j$
Expected arrival location of target j	$\bar{\boldsymbol{\rho}}_j$
Departure/arrival location uncertainty range of target j	$\tilde{\rho}_j$
Expected value of detecting target j	q_j

Table 3.2: **Assessed target data**

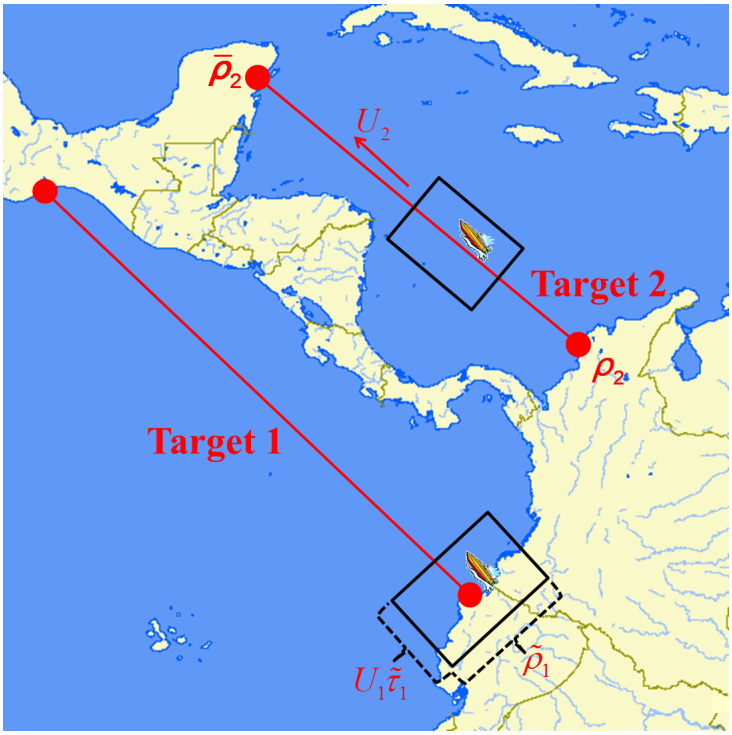


Figure 3.1: **Assessed target data illustration** - Assessed target data listed in Table 3.2 depicted for a two-target ($n = 2$) scenario. Expected departure time of targets (τ_j) and expected value of detecting targets (q_j) are not depicted.

We assume that targets are moving along straight-line *movement tracks* with constant speed, independent of the search effort. Modeling more complex target tracks is explored

in Section 4.2. We assume that the deviation from the expected departure time of target j is uniformly distributed within the range $[-\tilde{\tau}_j/2, \tilde{\tau}_j/2]$. Similarly, we assume that the deviation from the expected departure location of target j is uniformly distributed within the range $[-\tilde{\rho}_j/2, +\tilde{\rho}_j/2]$. This reflects the situation where search planners have information that allows them to bound departure time and departure location within fixed ranges, but lack either specific enough information or sufficient confidence in the available information to say any point in the range is more likely than another. Discrete-time and -space models are more flexible in this regard because, in principle, they can approximate many distributions by using a *heat map*, where each (potentially small) cell in the discretization may have a different probability that a target is contained within. Unfortunately, when planners lack high-fidelity target data, the heat map presents an illusion of precision. In real-world planning scenarios, given the dynamic nature of the mission and the complexity of the intelligence gathering process, a uniform distribution on uncertainty may often be the most appropriate.

Based on these data, the latest departure time for each target j can be calculated as

$$\tau_j^{min} = \tau_j + \frac{1}{2}\tilde{\tau}_j.$$

Similarly, the earliest arrival time for each target j can be calculated as

$$\tau_j^{max} = \tau_j - \frac{1}{2}\tilde{\tau}_j + \frac{1}{U_j}||\boldsymbol{\rho}_j - \bar{\boldsymbol{\rho}}_j||.$$

Lastly, we can calculate the velocity vector of target j, $\boldsymbol{u}_j$, as a function of speed, and expected departure and arrival locations.

We model search within each target's moving region of uncertainty (*search region*) using a random search law with known sensor sweep width W; for details on random search models see Washburn (2002, ch. 2) and Stone (1975, ch. 1). While, strictly speaking, random search is physically impossible, it serves as a useful model. In a real mission, the planner chooses the tactical pattern to be flown by the searcher in the search region. Unless the planner deliberately makes a poor choice, the search pattern will be better than random search. In that sense, random search provides a practical lower bound on detection probability.

We assume that $\tilde{\tau}_j$ and $\tilde{\rho}_j$ are small relative to the scenario time limit D and the dimensions

of the AOI, respectively. This assumption ensures that the search regions are not too large relative to the size of the AOI. A model where this assumption does not hold is explored in Section 4.4. We also assume that search actions cannot be conducted for more than one target at the same time. This reflects the operational setting where the searcher is seeking out a specific target in a sparse AOI looking for characteristics outlined in intelligence reports. Thus, in the unlikely event that search regions overlap in time and space, the searcher cannot receive additional reward for searching for more than one target at a time. We assume that the search involves a single searcher. A more general multi-searcher model is considered in Section 4.3. Given all the available information, the search controller wishes to route the searcher through the AOI in order to maximize the expected value of the search effort. We model this as routing a vehicle across a network $G = (N, A)$, where nodes are defined by the search regions and arcs are defined by the searcher's transit between each pair of moving search regions.

3.1 Formulation

Since targets are in linear motion, the distance required to travel directly between each pair (i, j) of search regions can be computed as a function of time using Euclidean distance calculations. We proceed under the assumption that the path of the searcher is through the respective center of each search region. This gives the planner the flexibility to dictate any tactical search pattern to the searcher by placing it in position to travel in any direction within the search region. We also assume that the error between Euclidean distance and great circle distance is small relative to the size of the search regions. This is a reasonable assumption to make in the near the equator, where this error is small, or when the entire AOI small. Suppose a searcher is searching for target in some predefined order and that the searcher has just completed searching region i. If a_i represents the time the searcher began searching region i and d_i represents the duration of the search in region i, then we can compute the current position $\boldsymbol{\psi}_i$ of the searcher as $\boldsymbol{\psi}_i = \boldsymbol{\rho}_i + (a_i + d_i - \tau_i)\boldsymbol{u}_i$. We assume that $a_i \geq \tau_i^{min}, \forall i$; the searcher will never arrive to search a target that has not departed. Suppose the searcher is next routed to region j, and that the transit time from region i to region j is denoted as $t_{i,j}$. The position of region j at the moment the searcher arrives is $\boldsymbol{\psi}_j = \boldsymbol{\rho}_j + (a_i + d_i + t_{i,j} - \tau_j)\boldsymbol{u}_j$. We can now relate the distance between region i and region j to the distance the searcher can travel in the same amount of time $||\boldsymbol{\psi}_i - \boldsymbol{\psi}_j|| = V t_{i,j}$. We can relax this relationship by recognizing that the searcher does not always have to travel at maximum cruise speed. The searcher could choose to

travel slower. Thus, $||\boldsymbol{\psi}_i - \boldsymbol{\psi}_j|| \leq V t_{i,j}$, which is second-order cone constraint in the time resources $\boldsymbol{a}, \boldsymbol{d}$, and $\boldsymbol{t}$ of the form (2.3b).

Drawing from search theory (Washburn 2002, ch. 2), we define the detection rate in search region j, α_j, as

$$\alpha_j = \frac{W\hat{V}}{\tilde{\tau}_j \tilde{\rho}_j U_j}. \tag{3.1}$$

We assume that the searcher speed $\hat{V}$ is much greater than the target speeds $U_j, \forall j \in \hat{N}$. From the searcher's perspective, within each search region, the target is essentially stationary. It is possible to model the problem where this does not hold (Washburn 2002, sec. 6-1). However, this assumption approximately holds in our SSP application, where we consider search aircraft and surface (e.g., boats) smugglers.

This problem can be formulated as the following MINLP, which is a special case of **P**.

Problem SSP:

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t},\boldsymbol{x},\boldsymbol{y}} \quad \sum_{j \in \hat{N}} q_j \left(1 - \exp\left\{-\alpha_j d_j y_j\right\}\right) \tag{3.2a}$$

$$\begin{aligned} \text{s.t.} \quad & (||\boldsymbol{\rho}_i + (a_i + d_i - \tau_i)\boldsymbol{u}_i - \boldsymbol{\rho}_j \\ & \quad \ldots - (a_i + d_i + t_{i,j} - \tau_j)\boldsymbol{u}_j|| \\ & \qquad \ldots - V t_{i,j}) x_{i,j} \leq 0, \qquad \forall (i,j) \in A \end{aligned} \tag{3.2b}$$

$$(a_i + d_i + t_{i,j} - a_j)\, x_{i,j} \leq 0, \qquad \forall (i,j) \in A \tag{3.2c}$$

$$\sum_{j \in \hat{N}} d_j + \sum_{(i,j) \in A} t_{i,j} \leq T \tag{3.2d}$$

$$\sum_{j \in N} d_j + \sum_{(i,j) \in A} t_{i,j} \leq D \tag{3.2e}$$

$$a_j \geq \tau_j^{min}, \quad \forall j \in N \tag{3.2f}$$

$$a_j + d_j \leq \tau_j^{max}, \quad \forall j \in N \tag{3.2g}$$

$$a_0 = 0 \tag{3.2h}$$

$$d_{n+1} = 0 \tag{3.2i}$$

$$a_j, d_j \geq 0, \qquad \forall j \in N \tag{3.2j}$$

$$t_{i,j} \geq 0, \qquad \forall (i,j) \in A \tag{3.2k}$$

$$\boldsymbol{y} = \boldsymbol{\Gamma x} \tag{3.2l}$$

$$\boldsymbol{x} \in \mathbb{X} \tag{3.2m}$$

The objective (3.2a) is to maximize the expected value of the search effort. Constraints (3.2b) ensure that search region j is reachable from search region i in time $t_{i,j}$ given the searcher's speed V. Constraints (3.2c) propagate arrival times $\boldsymbol{a}$ forward in time as arcs are traversed. Constraints (3.2b) and (3.2c) correspond to (2.3b) in **P**. Constraint (3.2d), corresponding to (2.3c) in **P**, ensures that the total flying hours of the searcher does not exceed its endurance limit T. Similarly, constraint (3.2e), corresponding to (2.3c) in **P**, ensures that the time horizon of plan does not exceed the scenario time limit D. Note that the left summation in (3.2d) is over the set of nodes not including the home station and recovery location $\hat{N}$. This may appear to be inconsistent with (2.3c), however in **SSP** we could equivalently model two d_j terms for each node. One retains the correct dwell resources at all nodes, and the other is nearly a copy but consumes zero dwell resources at nodes 0 and $n+1$. We choose the more compact formulation here. Constraints (3.2f) require that the vehicle be routed to search regions only after the target has surely departed. Similarly, constraints (3.2g) preclude searching in a region after the target has possibly arrived. Constraint (3.2h) ensures that the scenario starts at time 0, while constrain (3.2i) ensures that the scenario ends when the searcher arrives at the recovery location. Constraints (3.2f), (3.2g), (3.2h) and (3.2i) correspond to (2.3d) in **P**. Constraints (3.2j), (3.2k), (3.2l), and (3.2m) correspond to (2.3f), (2.3g), (2.3h), and (2.3i) in **P** respectively.

We define an NLP analogous to $\mathbf{P}(\boldsymbol{x})$ which, for any route $\boldsymbol{x} \in \mathbb{X}$, provides the optimal time resource expenditure. We arrive at this problem by fixing $\boldsymbol{x}$ and $\boldsymbol{y}$, and retaining from **SSP** only the interesting constraints and objective function terms. Recall that $N(\boldsymbol{x}) = \{j \in N : y_j = 1; y_0, = 1, \boldsymbol{y} = \boldsymbol{\Gamma x}\}$ and $A(\boldsymbol{x}) = \{(i,j) \in A : x_{i,j} = 1\}$. Additionally, we define the set of search regions in the path $\hat{N}(\boldsymbol{x}) = N(\boldsymbol{x}) \backslash \{0, n+1\}$.

Problem SSP($\boldsymbol{x}$):

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t}} \quad \sum_{j\in\hat{N}(\boldsymbol{x})} q_j\left(1-\exp\left\{-\alpha_j d_j\right\}\right) \tag{3.3a}$$

$$\text{s.t.} \quad ||\boldsymbol{\rho}_i + (a_i + d_i - \tau_i)\boldsymbol{u}_i - \boldsymbol{\rho}_j$$
$$\ldots - (a_i + d_i + t_{i,j} - \tau_j)\boldsymbol{u}_j||$$

$$\ldots - Vt_{i,j} \leq 0, \qquad \forall (i,j) \in A(\boldsymbol{x}) \tag{3.3b}$$

$$a_i + d_i + t_{i,j} - a_j \leq 0, \qquad \forall (i,j) \in A(\boldsymbol{x}) \tag{3.3c}$$

$$\sum_{j\in\hat{N}(\boldsymbol{x})} d_j + \sum_{(i,j)\in A(\boldsymbol{x})} t_{i,j} \leq T \tag{3.3d}$$

$$\sum_{j\in N(\boldsymbol{x})} d_j + \sum_{(i,j)\in A(\boldsymbol{x})} t_{i,j} \leq D \tag{3.3e}$$

$$a_j \geq \tau_j^{min}, \quad \forall j \in N(\boldsymbol{x}) \tag{3.3f}$$

$$a_j + d_j \leq \tau_j^{max}, \quad \forall j \in N(\boldsymbol{x}) \tag{3.3g}$$

$$a_j, d_j \geq 0, \qquad \forall j \in N(\boldsymbol{x}) \tag{3.3h}$$

$$t_{i,j} \geq 0, \qquad \forall (i,j) \in A(\boldsymbol{x}) \tag{3.3i}$$

(3.2h) and (3.2i)

We observe that **SSP($\boldsymbol{x}$)** is a model that can be used by planners, heuristics, and/or exact algorithms to evaluate the quality of any complete route $\boldsymbol{x}$. Some operational settings, perhaps based on a subjective prioritization, may call for planners to determine the order in which targets are searched. When this is the case the remaining decision problem is one of determining the search times $\boldsymbol{a}, \boldsymbol{d}$, and $\boldsymbol{t}$. This can be done quickly by solving the convex NLP **SSP($\boldsymbol{x}$)**.

After a Big-M reformulation, **SSP** can be equivalently stated as an MINLP with a convex continuous relaxation. Let $M_{i,j}^R$ be a number that is always greater than the distance between region i and region j. Let $M_{i,j}^T$ be a number that is always greater than the time required for the searcher to travel between region i and region j. Let M_j^D be a number that is always greater than the search time in region j. We arrive at the following Big-M-reformulated SSP.

Problem SSPM

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t},\boldsymbol{x},\boldsymbol{y}} \quad \sum_{j\in\hat{N}} q_j\left(1-\exp\left\{-\alpha_j d_j\right\}\right) \tag{3.4a}$$

$$\begin{aligned}
\text{s.t.} \quad & ||\boldsymbol{\rho}_i + (a_i + d_i - \tau_i)\boldsymbol{u}_i - \boldsymbol{\rho}_j \\
& \ldots - (a_i + d_i + t_{i,j} - \tau_j)\boldsymbol{u}_j|| \\
& \qquad \ldots - Vt_{i,j} \le M^R_{i,j}(1-x_{i,j}), \quad \forall (i,j)\in A \qquad (3.4b)\\
& \qquad a_i + d_i + t_{i,j} - a_j \le M^T_{i,j}(1-x_{i,j}), \quad \forall (i,j)\in A \qquad (3.4c)\\
& \qquad d_j \le M^D_j y_j, \quad \forall j\in N \qquad (3.4d)
\end{aligned}$$

$$\sum_{j\in\hat{N}} d_j + \sum_{(i,j)\in A} t_{i,j} \le T \tag{3.4e}$$

$$\sum_{j\in N} d_j + \sum_{(i,j)\in A} t_{i,j} \le D \tag{3.4f}$$

(3.2f) – (3.2m)

The nonlinear interactions between the binary variables and the continuous variables in **SSP** are modeled with Big-M terms on the right hand sides of (3.4b), (3.4c) and (3.4d). Constraint (3.4d) requires that search duration be zero when the corresponding search region is not visited, which makes it possible to remove the nonlinear interactions in the objective function (2.3a), yielding (3.4a). It is well known that unnecessarily large Big-M values lead to poor continuous relaxations and ultimately slow down computation time (Camm et al. 1990). In the case of the SSP, since target motion is linear, we can compute Big-M values based on the maximum distance between each pair of targets.

For any two target search regions i and j (home station possibly being one of them), the following convex NLP produces the minimum travel distance between them.

Problem D

$$\begin{aligned}
\delta^*_{i,j} = \min_{t} \quad & ||(\boldsymbol{\rho}_i + (t-\tau_i)\boldsymbol{u}_i) - (\boldsymbol{\rho}_j + (t-\tau_j)\boldsymbol{u}_j)|| \\
\text{s.t.} \quad & \max\{\tau_i^{min}, \tau_j^{min}\} \le t \le \min\{\tau_i^{max}, \tau_j^{max}\}
\end{aligned}$$

We let $\delta_{i,j} \equiv V^{-1}\delta^*_{i,j}$ be the minimum travel time resource expenditure between search region i and search region j. We proceed under the assumption that the home station and the recovery location are the same physical location, therefore $\delta_{0,j} = \delta_{j,n+1} = V^{-1}\delta^*_{0,j} = V^{-1}\delta^*_{j,n+1}$. This is usually the case in the type search planning problems we consider and it imposes no limitations on our model or solution procedures. The SSP instance of $\mathbf{RP}(\mathbf{0})$ is obtained when no path $\boldsymbol{x}$ is specified. We force $\boldsymbol{a}$ to take on lower bound values in order to allow $\boldsymbol{d}$ to take on highest possible values. When this is done $\boldsymbol{a}$ and $\boldsymbol{t}$ can be eliminated from the problem, resulting in the following NLP in the search time $\boldsymbol{d}$.

Problem RSSP(0)

$$\max_{\boldsymbol{d}} \quad \sum_{j\in\hat{N}} q_j \left(1 - \exp\left\{-\alpha_j d_j\right\}\right) \tag{3.6a}$$

$$\text{s.t.} \quad d_j \le \tau_j^{max} - \tau_j^{min}, \qquad \forall j \in N \tag{3.6b}$$

$$\sum_{j\in\hat{N}} d_j \le T - 2\min_{j\in N}\{\delta_{j,n+1}\} \tag{3.6c}$$

$$\sum_{j\in N} d_j \le D - 2\min_{j\in N}\{\delta_{j,n+1}\} \tag{3.6d}$$

$$d_{n+1} = 0 \tag{3.6e}$$

$$0 \le d_j \le \min\{T, D\} - 2\delta_{j,n+1}, \quad \forall j \in \hat{N} \tag{3.6f}$$

We observe the following for any search region j that is not in the current partial path. If it has an earliest arrival time τ_j^{max}, which is less than or equal to the latest departure time τ_i^{min} for any search region i that is in the current partial path, it should not be considered in extending the current partial path. Formally, we construct the set $N^{\times}(\hat{\boldsymbol{x}}_\ell) = \{j \in N\backslash N(\hat{\boldsymbol{x}}_\ell) : \tau_j^{max} \le \tau_i^{min}, i \in N(\hat{\boldsymbol{x}}_\ell)\}$. Analogous to (2.9), we denote by $\tilde{T}_j(\hat{\boldsymbol{x}}_\ell)$ the optimistic dwell time for search region j that is not in the partial path $\hat{\boldsymbol{x}}_\ell$.

$$\tilde{T}_j(\hat{\boldsymbol{x}}_\ell) = \max\left\{0, T - \delta_{\ell,j} - \delta_{j,n+1} - \sum_{(i,i')\in A(\hat{\boldsymbol{x}}_\ell)} \delta_{i,i'}\right\}$$

For any partial path $\hat{\boldsymbol{x}}_\ell$, we have the following relaxed NLP as a special case of $\mathbf{RP}(\hat{\boldsymbol{x}}_\ell)$.

Problem RSSP($\hat{\boldsymbol{x}}_\ell$)

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t}} \quad \sum_{j\in\hat{N}} q_j\left(1-\exp\left\{-\alpha_j d_j\right\}\right) \tag{3.7a}$$

$$\text{s.t.} \quad ||\boldsymbol{\rho}_i + (a_i + d_i - \tau_i)\boldsymbol{u}_i - \boldsymbol{\rho}_j$$
$$\ldots - (a_i + d_i + t_{i,j} - \tau_j)\boldsymbol{u}_j||$$
$$\ldots - V t_{i,j} \le 0, \qquad \forall (i,j) \in A(\hat{\boldsymbol{x}}_\ell) \tag{3.7b}$$
$$a_i + d_i + t_{i,j} - a_j \le 0, \qquad \forall (i,j) \in A(\hat{\boldsymbol{x}}_\ell) \tag{3.7c}$$
$$\sum_{j\in\hat{N}} d_j + \sum_{(i,j)\in A(\hat{\boldsymbol{x}}_\ell)} t_{i,j} \le T - (1-I_\ell)\delta_{\ell,n+1} \tag{3.7d}$$
$$\sum_{j\in N} d_j + \sum_{(i,j)\in A(\hat{\boldsymbol{x}}_{\boldsymbol{\ell}})} t_{i,j} \le D - (1-I_\ell)\delta_{\ell,n+1} \tag{3.7e}$$
$$d_j \le \tilde{T}_j(\hat{\boldsymbol{x}}_\ell), \qquad \forall j \in N\backslash N(\hat{\boldsymbol{x}}_\ell) \tag{3.7f}$$
$$I_\ell d_j = 0, \qquad \forall j \in N\backslash N(\hat{\boldsymbol{x}}_\ell) \tag{3.7g}$$
$$d_j = 0, \qquad \forall j \in N^{\times}(\hat{\boldsymbol{x}}_\ell) \tag{3.7h}$$
$$t_{i,j} \ge 0, \qquad \forall (i,j) \in A(\hat{\boldsymbol{x}}_\ell) \tag{3.7i}$$
$$\text{(3.2f) – (3.2j), (3.6b), and (3.6f)}$$

SSP can be solved by Algorithm B&B using **RSSP**($\hat{\boldsymbol{x}}_\ell$) relaxations and the lower bound initialization heuristics described next.

3.2 Heuristic Algorithms

Denoting the optimal solution to **SSP** as Z^*, we observe that if $\epsilon = 0$ and the initial guess $\boldsymbol{x}$ is provided to Algorithm B&B where $Z(\boldsymbol{x})^* = Z^*$, then the number of NLP solutions required to prove $\boldsymbol{x} = \boldsymbol{x}^*$ is constant regardless of how branching is done in step 2. This is a direct consequence of the fact that fathoming only depends on the lower bound. Of course this observation is not unique to our problem setting. In fact, it is true of any branch-and-bound algorithm provided the algorithm does not include more sophisticated fathoming rules. This observation is the main impetus to develop a reliable way of providing initial solutions to Algorithm B&B, possibly eliminating the need for complex branching strategies. Furthermore, when runtimes are too long in large problem

instances, efficient heuristics can be used to give operators a feasible search plan to follow while the optimal plan is being computed. In some cases runtimes may be so long that operators cannot afford to wait for an optimal plan to be computed, and a heuristic algorithm is the only viable option.

We now describe two heuristics. The first is a fast heuristic that imposes a fixed target search order, solving at most $2n + 2$ NLPs to compute a good search plan solution. The second heuristic computes good search plans by limiting the number of targets that may be assigned the searcher.

3.2.1 Fixed Order Heuristic

In order to provide a good initial solution to Algorithm B&B, we consider a five-phase *SSP heuristic* that relies on the knowledge that solving a GOP-RDR entails finding an acceptable balance between dwell and transit resource expenditure. Ramesh and Brown (Ramesh and Brown 1991) outline a four-phase heuristic for the TOP using a *bang-for-buck* ratio that relates the reward at each node to the bounds on transit time. We use a similar idea here, however since rewards and transit times are generally not known quantities, we consider a bang-for-buck ratio that relates expected search value to the area of the search region. We also add considerations for transit time by clustering targets based on temporal and spatial proximity. Throughout, we use $\mathbf{SSP}(\boldsymbol{x})$ to quickly determine the value of search plan at each iteration in the heuristic.

The SSP heuristic begins by defining Δ as the *temporal clustering parameter*, which controls how close we allow target clusters to be with respect to time. We assume that the problem instance of interest can be separated into spatial clusters $\sigma \in \Sigma$ based on geographical boundaries. This is the case in our SSP application where smugglers are transiting through water on either side of a large land mass. Furthermore, since we are concerned with seagoing smugglers, they cannot move from one region to another. We denote by K_σ the set of targets that belong to spatial cluster σ.

The SSP heuristic computes search plans by assigning all targets to clusters based on time and space proximity, fixing the order in which targets may be searched, and then removing low-valued targets. The five-phase SSP heuristic algorithm is stated as follows.

SSP heuristic:

Phase I (Initialization). Begin by solving the relaxed NLP **RSSP**(0). Permanently remove from consideration in the heuristic, all targets j such that $d_j = 0$ in the optimal solution. For each of the remaining targets, calculate the bang-for-buck ratios.

1. Initialize cluster count $k = 1$, order index $o = 1$, and null path $\boldsymbol{x} = \mathbf{0}$. Solve **RSSP**(0) and record the optimal solutions $\boldsymbol{d}^*$. Construct the set of searchable targets $\check{N} = N \backslash \{j \in N : d_j^* = 0\}$.
2. Compute bang-for-buck ratios
$$\beta_j = \frac{q_j}{\tilde{\tau}_j \tilde{\rho}_j U_j}. \tag{3.8}$$

Phase II (Target Clustering). Partition all remaining targets into spatial clusters. Next, further partition all targets into clusters of width Δ based on their earliest arrival times τ_j^{max}, forming K clusters $\kappa_k, k \in \{1, \ldots, K\}$. Order targets within each cluster κ_k in ascending order of τ_j^{max}. (In our SSP application, the searcher's home station is generally closer to target arrival locations than departure locations. Therefore it is more beneficial for search to take place at the end of the target's movement track than it is at the beginning. Earlier target arrivals represent search opportunities that vanish earliest in the mission execution period $[0, D]$. Thus, earlier arriving targets would likely be searched first if they are searched at all.) For each cluster, assign the value $\nu_k = \min\{\tau_j^{max} : j \in \kappa_k\}$. Order each cluster in ascending value ν_k. For all targets, assign orderings $\{O_j, j = 1, 2, \ldots\}$ from the first target in the first cluster through the last target in the last cluster. Form the initial path $\boldsymbol{x}$ and solve **SSP**($\boldsymbol{x}$). If the problem is feasible, set the value Z_H^* to the optimal objective function value of this problem. Otherwise, set $Z_H^* = -\infty$.

3. For each spatial cluster $\sigma \in \Sigma$:

   Initialize time window parameter
$$\check{\tau}_\sigma = \min_{j \in \check{N} \cap K_\sigma} \{\tau_j^{max}\}.$$
   While
$$\check{\tau}_\sigma + \Delta < \max_{j \in \check{N} \cap K_\sigma} \{\tau_j^{max}\} :$$
   For each target $j \in K_\sigma$:

Assign target j to cluster κ_k if $\tau_j^{max} \in [\check{\tau}_\sigma, \check{\tau}_\sigma + \Delta)$.

Increment $\check{\tau}_\sigma$ to $\check{\tau}_\sigma + \Delta$.

If one or more targets are assigned to cluster κ_k in this time interval, increment k to $k+1$.

4. For each cluster κ_k:

   Order targets $j \in \kappa_k$ in ascending value of τ_j^{max}. Compute cluster order value $\nu_k = \min\{\tau_j^{max} : j \in \kappa_k\}$.

5. For each cluster κ_k, considered in ascending order value ν_k:

   For each target $j \in \kappa_k$:

   Assign search order $O_j = o$ and increment o to $o+1$.

6. Assign order $O_0 = 0$ to the home station and order $O_{n+1} = |\check{N}| + 1$ to the recovery location.

7. Form the initial path $\boldsymbol{x}$ by setting $x_{i,j} = 1$ for all i and j with consecutive orderings O_i and O_j. Solve **SSP**($\boldsymbol{x}$). Save incumbent path $\bar{\boldsymbol{x}} = \boldsymbol{x}$. If the problem is feasible, save the heuristic objective value Z_H^* as the optimal objective function value of this problem. Otherwise, set $Z_H^* = -\infty$.

Phase III (Feasibility Check). (Perform only if $Z_H^* = -\infty$.) Attempt to find an initial feasible solution by removing the target with the smallest value β_j from the route. When a target j is removed, generate an arc in the route between the target with order O_{j-1} and order O_{j+1}, then solve **SSP**($\boldsymbol{x}$). If a feasible solution is found, save the path and move to the next phase. Otherwise, continue this procedure removing one target at a time until an initial feasible solution is found. Set Z_H^* to the objective function value of this solution.

8. If $Z_H^* = -\infty$:

   For all targets $j \in \check{N}$, considered in ascending order β_j:

   Do **procedure** *Remove_j*: { Remove target j from the path $\boldsymbol{x}$ by setting $x_{i,j} = 0$ (for $i : O_i = O_j - 1$), $x_{j,i'} = 0$ (for $i' : O_{i'} = O_j + 1$), and $x_{i,i'} = 1$ (for $i : O_i = O_j - 1$ and $i' : O_{i'} = O_j + 1$). Remove j from the set $\check{N}$. Solve **SSP**($\boldsymbol{x}$). }

   If a feasible solution is found, set Z_H^* to the objective function value of this solution and go to step 9.

Phase IV (Cluster Seam Refinement). For each cluster in the search order, do the following seam refinement. Let target i be the last target in cluster k. Let target j be the first target in cluster $k+1$. If $\tau_i^{max} > \tau_j^{max}$ and $\beta_i < \beta_j$, temporarily remove target i from the route and form a new path by generating an arc between target j and the target with order O_{i-1}. If Z_H^* is improved, permanently remove target i from consideration in the heuristic. Otherwise, reinsert target i in its place in the route.

9. Save incumbent path $\bar{\boldsymbol{x}} = \boldsymbol{x}$. For each seam between clusters κ_{k-1} and κ_k, where $k > 1$:

   Let j be the target in that last order position in cluster κ_{k-1}. Let i' be the target in the first position in cluster κ_k. If $\tau_j^{max} > \tau_{i'}^{max}$ and $\beta_j < \beta_{i'}$, do *Remove_j* defined in step 8. If Z_H^* is improved, save incumbent path $\bar{\boldsymbol{x}} = \boldsymbol{x}$. Otherwise, reset incumbent path $\boldsymbol{x} = \bar{\boldsymbol{x}}$.

Phase V (Greedy Target Removal). Attempt to improve the solution by removing the target with the smallest value β_j from the route, generating a new route as described in Phases III and IV. If the solution is improved, permanently remove target j from consideration in the heuristic. Otherwise, reinsert target j in its place in the route. Consider all remaining targets in ascending order of β_j for removal in turn. Return the best route found upon completion of the above steps.

10. For each target $j \in \check{N}$, considered in ascending order β_j:

    Do *Remove_j* defined in step 8. Solve $\mathbf{SSP}(\boldsymbol{x})$. If Z_H^* is improved, save incumbent path $\bar{\boldsymbol{x}} = \boldsymbol{x}$. Otherwise, reset incumbent path $\boldsymbol{x} = \bar{\boldsymbol{x}}$.
11. Return heuristic path $\bar{\boldsymbol{x}}$ and solution Z_H^*.

The worst case run-time of this heuristic is $2n + 2$ NLP solutions. This occurs when a feasible route is found in Phase II, and all targets are considered for removal in Phases IV and V. In this situation each target occupies its own cluster. This can be prevented in well posed problem instances where Δ is chosen appropriately with respect to the arrival times τ_j^{max}. Since the NLP subproblems, $\mathbf{SSP}(\boldsymbol{x})$, can be solved quickly, approximately 1/10 of a second for problem instances with up to 20 targets, the heuristic is quite fast even in the worst case.

The bang-for-buck parameter (3.8) can be strengthened to account for targets of different

type by incorporating sweep with.

$$\hat{\beta}_j = \alpha_j q_j \tag{3.9}$$

The expression (3.9) simply captures the initial *rate-of-reward* garnered by searching each target j. When the sweep widths associated with the various targets are the same (3.8) and (3.9) clearly yield the same search order.

3.2.2 Partial Branch-and-Bound Heuristic

Another heuristic, the *PBB heuristic*, is based on exploring a limited portion of the B&B enumeration tree. Formally, we use Algorithm B&B and limit the level l in step 2 to some number $\bar{l}$, where $0 < \bar{l} < n + 1$. By an $\bar{l}$-level PBB heuristic we mean an implementation of Algorithm B&B that is restricted to exploring only the first $\bar{l}$ levels of the enumeration tree. This approach can be described operationally as restricting the search to plans that consider searching no more than $\bar{l} - 1$ targets.

This heuristic has several operational benefits. First, it may be the case that planners do not want to assign searchers to hunt for too many targets. There may be several reasons for this. When cruise (transit) altitude is different from patrol (search) altitude, efficiency is lost with repeated changes in altitude. If too many targets are in the plan the searcher's endurance may be diminished. Sensors may be tuned or aircrew may be conditioned to look for a specific type of target vessel, and having to search for many different types of targets (e.g., GO FAST, SPSS, merchant vessel, etc.) in the same mission may yield some subtle loss in efficiency. Second, using the PBB heuristic does not preclude planners from continuing to run Algorithm B&B in search of an optimal solution. One can think of a PBB heuristic solution as an intermediate result. Third, since Algorithm B&B is being used, the PBB heuristic can return meaningful optimality gap bounds while the SSP heuristic cannot.

3.3 SSP Numerical Experiments

We consider **SSP** where smugglers move through an abstract AOI, which is similar to Central America but with simplified coastal features, in a northwesterly direction as they attempt to transport illicit material from the south. In this scenario, we assume smuggler movement occurs through corridors defined by linear coastal strips of likely departure and arrival locations as depicted by the map shown in Figure 3.2. We observe that, even though the dotted lines intersect on the spatial map, target search regions rarely overlap

in space and time.

We assume searcher and target performance data that is consistent with known planning factors for P-3 aircraft and GO FAST smuggler boats; see Tables 3.3 through 3.5. We assume further that mission planning occurs in a 24-hour cycle, $D = 24$ hours. Departure time uncertainty data is randomly generated within the mission planning period with uncertainty ranging up to four hours. Location uncertainty data is randomly generated where smugglers are equally likely to depart and arrive anywhere on the aforementioned coastal strips. Expected value of detecting each target is randomly generated within the uniform range $[500, 5000]$ lbs, corresponding to estimated payload capacity of GO FAST boats.

Searcher	V	$\hat{V}$	T	D	W	Home
P-3	325	205	10	24	15	(6.5 8.0)

Table 3.3: **Searcher data** - Used throughout Section 3.3. Home station location is given in 100 nautical mile units with $(h\ v)$ being respective horizontal and vertical displacement from the origin depicted in Figure 3.2.

Target	U	τ	$\tilde{\tau}$	$\tilde{\rho}$	q
GO FAST	[55,65]	[0,12]	[1,4]	[20,100]	[500,5000]

Table 3.4: **Uniform target data ranges** - Used throughout Section 3.3

Corridor	$\boldsymbol{\rho}$	$\bar{\boldsymbol{\rho}}$
BL	[(12.0 0.0), (13.8 3.0)]	[(3.0 9.6), (7.2 7.8)]
C	[(13.8 4.8), (16.8 7.2)]	[(8.4 9.6), (10.2 9.0)]
TR	[(13.8 4.8), (16.8 7.2)]	[(7.2 10.8), (7.8 12.6)]

Table 3.5: **Target departure and arrival location corridors** - Used throughout Section 3.3. BL, C, and TR refer to the bottom-left, center, and top-right corridors respectively in Figure 3.2.

We solve 100 randomly generated problem instances with 3, 5, 7, 8, and 10 smugglers and compare model performance using Algorithm B&B with SSP heuristic initialization applied to **SSP** versus solving **SSPM** directly using two MINLP solvers. For each set of problem instances we deem the *best solver* to be the one that identifies an optimal solution in the shortest amount of average computing time. For the purposes of the numerical experiment, Algorithm B&B is implemented with a depth-first-search strategy and the optimality tolerance of zero, $\epsilon = 0$. We also omit constraints (3.7f) and (3.7h) from

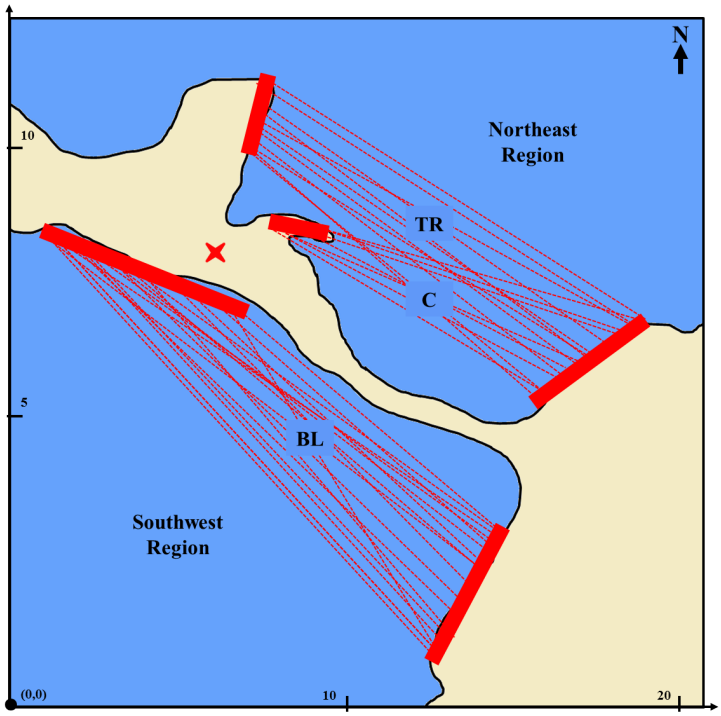


Figure 3.2: **Target track sample space** - Target movement tracks are randomly generated with origin and destination points chosen within coastal strips marked by thick solid bars. Given an origin and destination, the target track goes along a straight line within corridors indicated by dotted lines in a northwesterly direction. The searcher's home station is identified by "×". Dotted lines shown illustrate possible target movement tracks. Randomly generated target movement tracks are not limited to those depicted here, but stay within the envelope boundaries. Vertical and horizontal axes are in units of 100 nautical miles. Refer to Table 3.5 for listing of corridor endpoints.

RSSP($\hat{\boldsymbol{x}}_\ell$). This is done for the following two reasons. First, in initial testing, constraint (3.7f) rarely yielded fewer NLP solutions, made each NLP slightly more difficult to solve, and produced a slight increase in overall runtime. Second, the expected departure time range given in Table 3.4 rarely produces nonoverlapping arrival time feasibility windows $[\tau^{min}, \tau^{max}]$. Therefore, using these constraints in this experiment is not beneficial. The SSP heuristic is implemented with temporal clustering parameter $\Delta = 6$ hours and spatial clustering corresponding to the Southwest and Northeast regions. All computations are done on a 64-bit Windows 7 desktop computer (2x Intel Xeon 3.46GHz; RAM 24GB) using GAMS 23.8. We use MINOS to solve all NLP subproblems. In initial testing, DICOPT and BONMIN with ECP solver option appeared to be the most effective GAMS-based solvers for directly solving **SSPM**. Accordingly we limit our MINLP numerical results to these two solvers. For brevity, we refer to Algorithm B&B with heuristic

initialization as B&B and BONMIN with ECP solver option as BONMIN(ECP). For the remainder of this chapter, unless otherwise stated, searcher and target data is presented in nautical miles, nautical miles per hour, hours, and pounds; computation runtimes are given in seconds; and optimality gaps are reported as a percent difference from the optimal objective function value.

As indicated in Table 3.6, all three solvers are able to solve all 100 of the 5-target SSPs to optimality within 13 seconds computing time. For these problem instances BONMIN(ECP) is the best solver, while B&B yields the slowest mean runtime.

	B&B	BONMIN(ECP)	DICOPT
Num Solved	100	100	100
Runtime (sec)			
Mean	5.77	1.25	2.03
Std Dev	2.52	0.30	0.89
Std Error	0.25	0.03	0.09
Median	5.44	1.22	1.92
90th Percentile	9.21	1.70	3.08
Min	1.54	0.61	0.61
Max	13.00	1.98	4.99

Table 3.6: **Runtime summary for 5-target SSPs** - Num Solved refers to the number of problems out of 100 that were solved to optimality within 30 minutes. BONMIN(ECP) dominates in all metric categories and B&B appears to be the least favorable.

We use *performance profiles* (Dolan and Moré 2002) as a method for comparing solver runtimes. Performance profiles require two components: *performance ratios* and *performance metrics*. A performance ratio is a ratio of the runtime for solver s on problem p to the best runtime for all solvers tested on problem p. A performance metric is the empirical probability, across all problems p, that the runtime for solver s is within a factor of k of the best solver runtime. A performance profile is a distribution function of the performance metric over factors k.

We see in Figure 3.3 that BONMIN(ECP) preforms well on 5-target SSPs, with the fastest runtime for nearly 90% of these problems. DICOPT runtimes stay within a factor of three of the fastest runtime for 90% of problems. B&B lags behind the MINLP solvers, with runtimes within a factor of seven of the fastest runtimes for approximately 80% of problems. All of the problem instances being examined here are solved in 13 seconds or less. On a relative (performance profile) scale BONMIN(ECP) seems to be the clear

winner, but all of these solvers would be acceptable to planners in a practical sense.

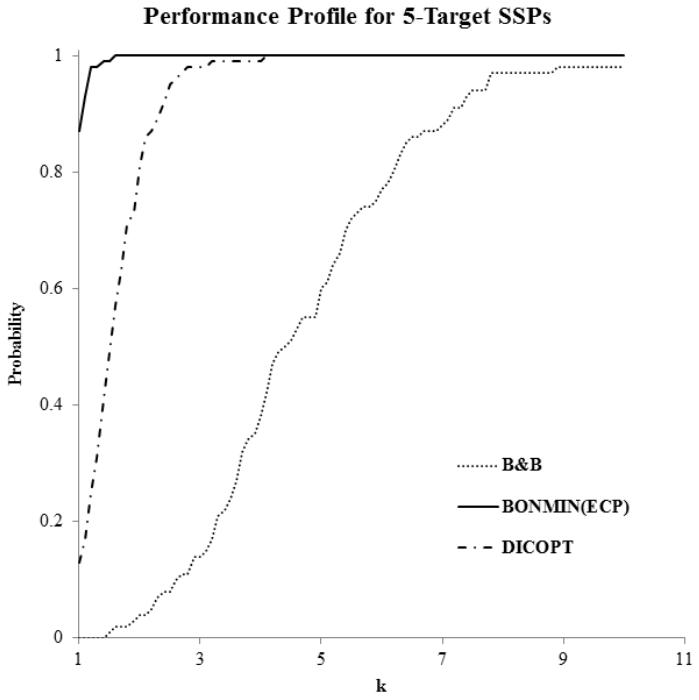


Figure 3.3: **Performance profile for 5-target SSPs** - BONMIN(ECP) preforms well on 5-target SSPs, with the fastest runtime for nearly 90% of these problems. B&B lags behind the MINLP solvers, with runtimes within a factor of seven of the fastest runtimes for approximately 80% of problems.

On the 7- and 8-target SSPs the relative performance of these solvers changes dramatically. Table 3.7 and the performance profile plot, Figure 3.4, highlight that runtimes of BONMIN(ECP) and B&B on 7-target SSP instances are nearly identical. DICOPT yields the slowest runtimes of the three solvers tested.

Table 3.8 shows that B&B is the best solver for the larger 8-target problem instances. B&B runtimes are, on average, 90 seconds (1.5 minutes), while BONMIN(ECP) runtimes are much larger at 267 seconds (4.5 minutes). We observe that since the limiting distribution of the sample mean is normal, and considering the standard error of the mean runtimes, we can say with high ($> 99\%$) confidence that the true mean runtimes for B&B on all 8-target problems in this sample space are faster than that of the other two solvers.

The performance profile plot (Figure 3.5) demonstrates that runtimes for BONMIN(ECP) and DICOPT in nearly all problem instances are several times larger than that of B&B,

	B&B	BONMIN(ECP)	DICOPT
Num Solved	100	100	100
Runtime (sec)			
Mean	32.84	30.47	115.39
Std Dev	20.48	16.31	97.76
Std Error	2.05	1.63	9.78
Median	27.82	26.19	86.20
90th Percentile	53.06	52.99	240.25
Min	8.77	8.02	15.90
Max	141.07	81.96	618.70

Table 3.7: **Runtime summary for 7-target SSPs** -BONMIN(ECP) and B&B have nearly identical runtimes. DICOPT yields the slowest runtimes of the three solvers tested.

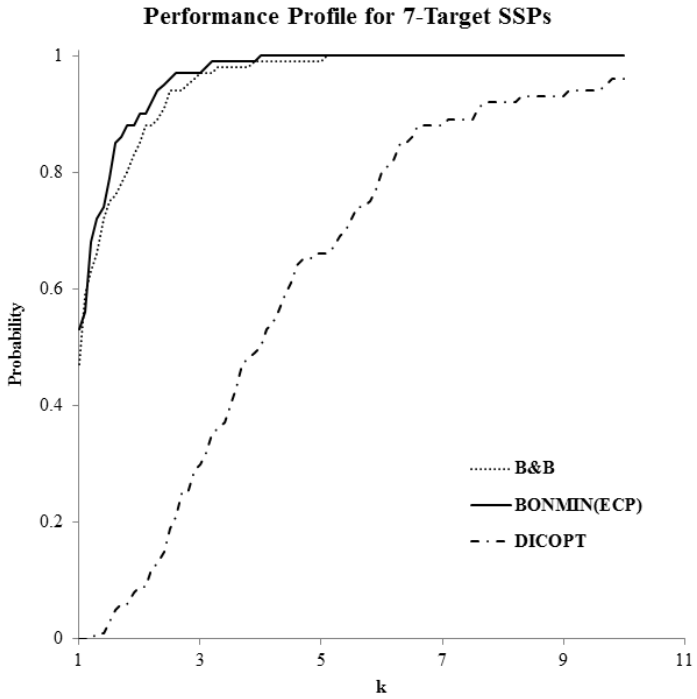


Figure 3.4: **Performance profile for 7-target SSPs** - BONMIN(ECP) and B&B each have the best runtimes in approximately half of the test problems. DICOPT yields the slowest runtimes of the three solvers tested.

with over half of the probability mass for BONMIN(ECP) being in the $k = 3$ to $k = 9$ range.

When considering larger, 10-target, SSPs it is clear that B&B is the only viable algorithm among the three tested. Table 3.9 highlights that B&B is able to solve 97 of 100 prob-

	B&B	BONMIN(ECP)	DICOPT
Num Solved	100	100	82
Runtime (sec)			
Mean	90.01	267.82	876.74
Std Dev	72.13	185.82	578.37
Std Error	7.21	18.58	57.84
Median	67.38	191.67	737.59
90th Percentile	181.18	500.62	1800.00
Min	8.74	65.58	57.69
Max	415.33	946.74	1800.00

Table 3.8: **Runtime summary for 8-target SSPs** - B&B is the best solver for these problem instances. B&B runtimes are, on average, 90 seconds (1.5 minutes), while BONMIN(ECP) runtimes are much larger at 267 seconds (4.5 minutes).

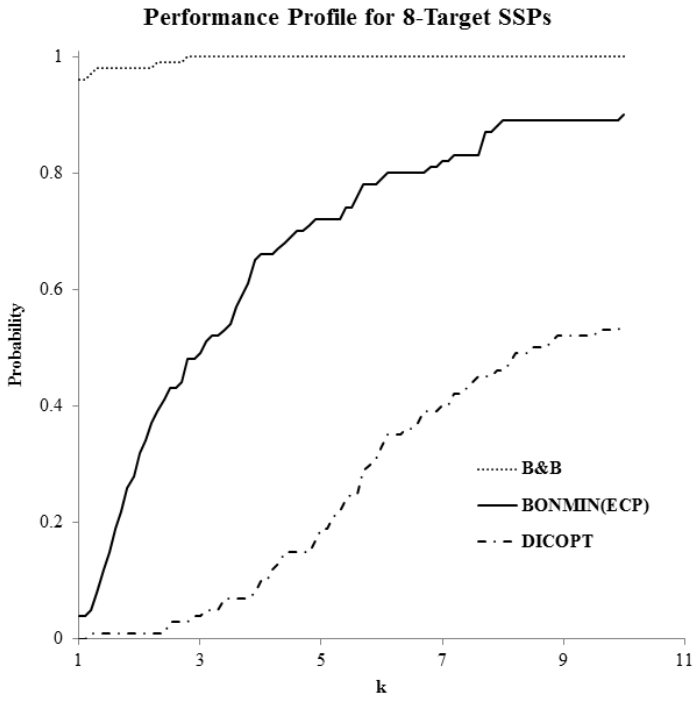


Figure 3.5: **Performance profile for 8-target SSPs** - B&B yields the fastest runtimes for nearly all problem instances. BONMIN(ECP) and DICOPT runtimes are at least three times larger than that of B&B for over 60% of problems tested.

lem instances within 30 minutes of computing time. The mean runtime is 8.5 minutes and 90% of the problems are solved within 17 minutes of computing time. Meanwhile BONMIN(ECP) and DICOPT are unable to solve any of the SSP test problems within

	B&B	BONMIN(ECP)	DICOPT
Num Solved	97	0	0
Runtime (sec)			
Mean	515.91	-	-
Std Dev	697.87	-	-
Std Error	69.79	-	-
Median	313.06	-	-
90th Percentile	1006.96	-	-
Min	33.27	-	-
Max	1800.00	-	-

Table 3.9: **Runtime summary for 10-target SSPs** - B&B is the only viable solver for these problems. BONMIN(ECP) and DICOPT are unable to solve any of these problems within 30 minutes of computing time.

30 minutes.

We now quantify how far the BONMIN(ECP) and DICOPT solutions are from the 10-target SSP optimal solutions by looking at the reported optimality gaps upon termination. Table 3.10 highlights that when BONMIN(ECP) and DICOPT terminate upon reaching the 30 minute time limit, the solution available is usually far from optimal. On average, solutions are off by a factor of at least 3.5 (optimality gaps in excess of 350%). For over half of the problems tested, DICOPT is unable to provide a bound on the optimal solution because the initial MIP for the linearized subproblem is not solved within 30 minutes.

The trend continues for larger problems. On a set of 25 randomly generated 15-target problem instances B&B solves each problem instance to optimality in 4,667.8 seconds (1.30 hours) on average, solving 18 out of 25 problem instances within 2 hours. BONMIN(ECP) is unable to solve any of these problem instances within 2 hours, terminating with an average optimality gap of 949%.

We observe that, on average, 5% of the B&B runtime is spent by MINOS solving the partial path relaxation problems $\mathbf{RSSP}(\hat{\boldsymbol{x}}_\ell)$. The majority of the total runtime is spent generating these problems within GAMS and managing the enumeration tree. It seems reasonable to conclude that a more efficient implementation of B&B would yield substantially shorter runtimes.

We are able to gain some insight into why B&B outperforms the MINLP solvers as

	BONMIN(ECP)	DICOPT
Num Bounds	100	44
Gap (%)		
Mean	504	358
Std Dev	323	223
Median	433	316
10th Percentile	236	126
90th Percentile	903	551
Min	74	17
Max	1893	1182

Table 3.10: **Optimality gap summary for BONMIN(ECP) and DICOPT on 10-target SSPs** - Num Bounds refers to the number of problem instances out of 100 for which the respective solver provided a bound on the objective function value within 30 minutes of computing time. Reported solutions generally differ from the optimal solutions by a large margin, upwards of 350% for both solvers. DICOPT is unable to report an optimality gap within 30 minutes for 56 problem instances.

the problem size increases by examining the branch-and-bound enumeration tree of a representative problem instance. We consider a 10-target SSP instance that is solved in 324 seconds, near the median runtime. In order to isolate the efficiency gained by the heuristic, we solve this problem with no initial solution provided as well as with heuristic initialization. We note that a 10-target SSP results in an enumeration tree of nearly 20 million nodes, spanning 12 levels deep. At each node we solve $\mathbf{RSSP}(\hat{\boldsymbol{x}}_\ell)$. Clearly, B&B visits only a small fraction of these nodes due to fathoming. Any path through the tree that has length 12 is a path that visits all target search regions. We use the term *perceived depth* to refer to the depth of visited nodes in the enumeration tree. If the average perceived depth of the tree were large, it would be tantamount to enumerating all possible paths $\boldsymbol{x} \in \boldsymbol{X}$. Thus, in order for our B&B algorithm to perform efficiently we need that the perceived depth of the tree remain relatively small for large problems. This is possible due to the dwell-to-transit resource trade-off that takes place when we consider extensions to partial paths in the enumeration tree. Table 3.11 shows the number of nodes at each level of the tree for a representative problem instance with and without heuristic initialization. We see that the tree is explored no more than 9 levels deep, as the partial path relaxation provided by $\mathbf{RSSP}(\hat{\boldsymbol{x}}_\ell)$ encounters the optimal solution bound fairly shallow in the enumeration tree. The majority of the nodes are visited in levels 5-7 of the enumeration tree. Considering that a 5-target SSP, solvable in only a few seconds, yields an enumeration tree that is seven nodes deep, the perceived depth of the tree for

larger 10-target SSP is shallow relative to its problem size.

Level	With SSP Heuristic # NLPs Solved	Without SSP Heuristic # NLP Solved
1	1	1
2	10	10
3	100	100
4	657	702
5	2008	3024
6	2051	5061
7	714	3354
8	85	720
9	0	40
Total	5626	13 012
Runtime (sec)	324.78	676.69

Table 3.11: **Number of NLP solved on each level of the B&B tree for a representative 10-target SSP instance** - The perceived depth of the enumeration tree is shallow relative to its problem size, highlighting the resource trade-off that motivates the GOP-RDR and highlights the benefit of the Algorithm B&B.

Table 3.11 also shows the benefit of using the SSP heuristic to determine the initial guess for B&B. Runtime increases proportional to the number of required NLP solutions in the enumeration tree. Having a good bound on the optimal solution reduces the total number of required NLP solutions by a factor of 2.3.

We conclude this section with some remarks on the performance of the heuristics presented in Section 3.2 with respect to the 3-, 5-, 7-, 8-, and 10-target SSP test set. The SSP heuristic correctly identifies 223 optimal solutions out of 500 total SSPs tested. Table 3.12 shows that the accuracy of this heuristic tends to diminish as the number of targets increases. However, the average relative optimality gap remains within the 1-3% range throughout. Therefore, while the accuracy rate in finding the optimal solution decreases, the SSP heuristic does not miss by too wide of a margin on average. This heuristic is able to get within 7% of optimality in at least 90% of all problems tested. In all cases, the average accuracy rate is driven by one to three poor performing problem instances. While the average accuracy diminishes, the runtime remains fairly constant. It is at or below half a second for all problems tested. This is consistent with the worst case runtime analysis presented in Section 3.2.1. We observe that, comparing Table 3.12 to Table 3.9, on average the heuristic's 90th percentile optimality gaps for 10-target problem instances are smaller than the optimality gaps for BONMIN(ECP) and DICOPT. A problem-by-

problem comparison reveals that the heuristic solutions have a smaller optimality gap in all problem instances.

	Number of Targets				
	3	5	7	8	10
Avg Time (sec)	0.19	0.24	0.30	0.31	0.34
Min Time (sec)	0.14	0.17	0.22	0.19	0.22
Max Time (sec)	0.40	0.42	0.65	0.55	0.67
Num Optimal	80	42	41	31	29
Avg Gap (%)	1	3	2	3	2
90th Percentile Gap (%)	3	7	6	6	6
Max Gap (%)	29	34	23	48	30

Table 3.12: **SSP heuristic performance results** - The heuristic presented in Section 3.2.1 produces runtimes that are at or below the half-second mark for all 3-, 5-, 7-, 8-, and 10-target SSP instances. Accuracy in finding the optimal solution appears to diminish as problem size increases, however the average, 90th percentile, and maximum optimality gaps remain fairly stable.

Numerical results of the PBB heuristic highlight its utility in producing optimality gap estimates. While the SSP heuristic demonstrates a stable gap between its solution and the true optimal solution, the true solution will not be known in an arbitrary problem instance. Therefore, having the ability to provide an estimate of the optimality gap is desirable so that planners can be assured that a plan based on a purely heuristic solution is good by some measure.

In all 500 sample problems, the solution returned by the PBB heuristic is the same as the SSP heuristic solution, thus the accuracy and true (unknown) optimality gap performance of these heuristics are the same. While this might suggest that the two heuristics always produce the same solution, this is not true in general. We consider a two-target scenario where one target moves away past the home station and the other target moves toward the home station. The target data for this example scenario is provided in Table 3.13. By virture of the smallest τ^{max} value the SSP heuristic produces a plan that searches target 2 first and target 1 last, yielding an objective function with expected search value 1,583.9. In this situation it is clearly more beneficial to use the opposite search order to decrease total transit time as target 1 moves away from the home station. The optimal search plan, swapping the targets in the search order, yields an objective function with expected search value 1,743.7. In this example the PBB heuristic solution differs from the SSP heuristic solution.

Target	U	τ	$\tilde{\tau}$	τ^{min}	τ^{max}	$\boldsymbol{\rho}$	$\bar{\boldsymbol{\rho}}$	$\tilde{\rho}$	q
1	60	3	2	4	23.1	(13.8 3.0)	(3.0 9.6)	50	1000
2	60	9	2	10	22.4	(13.8 3.0)	(6.6 7.8)	50	1000

Table 3.13: **2-target example scenario data**

Table 3.14 highlights that the runtimes for the 3-level PBB heuristic on 3- and 5-target problem instances are, as one would expect, similar to the runtimes for the full B&B. However, for the larger 7-, 8-, and 10-target problem instances running the PBB heuristic takes a fraction of the time it takes to obtain the optimal solution using Algorithm B&B. The optimality gap estimate produced by the PBB heuristic remains fairly stable as problem size (e.g., number of targets) increases. While the pure accuracy of solution obtained by either heuristic diminishes as the problems get larger, the true (unknown) and the estimated optimality gaps appear to be stable.

	Number of Targets				
	3	5	7	8	10
Avg Time (sec)	0.83	3.77	11.64	18.55	37.92
Full B&B Avg Time (sec)	0.83	5.77	32.84	90.01	515.91
Num Optimal	80	42	41	31	29
Avg Gap Est (%)	1	14	13	14	14
90th Percentile Gap Est (%)	4	27	25	22	23
Max Gap Est (%)	14	68	44	41	33

Table 3.14: **3-level PBB heuristic performance results** - The heuristic presented in Section 3.2.2 produces solutions in a fraction of the time it takes to run the full B&B algorithm on moderate to large problem instances. In all 500 sample problems, the solution returned by the PBB heuristic is the same as the SSP heuristic solution, thus the accuracy and true (unknown) optimality gap performance of these heuristics are the same. The PBB heuristic yield average, 90th percentile, and maximum optimality gap estimates remain fairly stable as problem size increases.

CHAPTER 4:
Search Model Enhancements

Problem **SSP** serves as a baseline model for solving many interesting problems that arise in search planning. Real-world scenarios may require developing search plans for multiple searchers, coordinating surface interdiction efforts, and accounting for complicated smuggler movement tracks over multiple planning periods. This chapter demonstrates how **SSP** can be enhanced to capture these difficult planning issues.

4.1 Benchmark Scenario

In order to motivate the search model enhancements presented later in this chapter, we describe the following benchmark scenario. This scenario was developed so that it is representative of the real-world issues faced by planners, while providing scenario details in an unclassified manner. Our benchmark scenario (**BS**) involves nine targets, two aerial searchers, and four surface interdiction assets (*interdictors*). Figure 4.1 depicts a spatial representation of the expected movement tracks for these targets.

There are two P-3 aircraft available for planning, each operating out of Comalapa Airbase, El Salvador. Each has a maximum cruise speed (V) of 325 knots, an on-station speed ($\hat{V}$) of 205 knots, and a 10-hour endurance (T). We assume a revolving daily planning cycle and a total scenario time limit (D) of 72 hours. In order to minimize disruption to the daily planning cycle rhythm, we require that search planning recommendations be computed within a *reasonable amount* of time. We define a reasonable amount of time to be two hours. Runtimes in excess of two hours are deemed to be too long because having planners wait for these solutions to be computed may negatively impact other aspects of the planning process.

Throughout this scenario, there are four surface ships available for interdiction support as needed. Two can be positioned anywhere in the eastern Pacific Ocean and the other two can be positioned anywhere in the Caribbean Sea. Each has an intercept speed of 20 knots. Surface interdictors are to be positioned in coordination with the search plan so that they are able to respond if a searcher detects a target. When a target is detected by a searcher, the searcher monitors the target and summons an interdictor. The interdictor must then transit from its standby location to the location where the target is being

monitored. A successful interdiction occurs once the interdictor arrives. Due to high cost of repositioning surface assets, we assume that interdictors cannot be routed and that they will remain in their stand-by position until called upon to respond when a detection occurs.

The first target (GF1) is a GO FAST boat that is expected to be carrying 1,000 kg of cocaine and traveling in the Caribbean at an expected speed (U_1) of 50 knots. The expected track of GF1 is a coastal route that transits through the longitude/latitude points: 76.3W 9N – 79W 10N – 82W 10N – 83.5W 14N. The location of departure (76.3W 9N) and the arrival location (83.5W 14N) both have a uniform uncertainty range of ±30 nm. There is a 60 nm × 60 nm uniform range of uncertainty about each waypoint (79W 10N, 82W 10N). GF1 is expected to depart at 0000 hours on day one of the scenario with uniform departure time uncertainty of ±2 hours. The overall certainty of the intelligence about GF1 is 0.95, thus the expected detection value (q_1) of GF1 is the product of the payload and the overall certainty $q_1 = 1,000 \times 0.95 = 950$. The sweep width of a P-3 against a GO FAST boat is assumed to be 15 nm.

The second target (SP1) is a Self-Propelled Semi-Submersible vessel (SPSS) that is expected to be carrying 5,000 kg of cocaine and traveling in the Caribbean at an expected speed (U_2) of 15 knots. The expected track of SP1 is a straight-line route that transits through the longitude/latitude points: 76.3W 9N – 83.5W 14N. The location of departure (76.3W 9N) and the arrival location (83.5W 14N) both have a uniform uncertainty range of ±30 nm. SP1 is expected to depart at 0400 hours on day two of the scenario with uniform departure time uncertainty of ±4 hours. The overall certainty of the intelligence about SP1 is 0.95, thus the expected detection value (q_2) of SP1 is the product of the payload and the overall certainty $q_2 = 5,000 \times 0.95 = 4,750$. Due to the low profile of an SPSS, the sweep width of a P-3 against this type of vessel is assumed to be 5 nm. By virtue of its large expected detection value and its relatively narrow uncertainty range, SP1 is considered a high-value target in this scenario.

The third target (GF2) is a GO FAST boat that is expected to be carrying 1,500 kg of cocaine and traveling in the Caribbean at an expected speed (U_3) of 50 knots. The expected track of GF2 is a straight-line route that transits through the longitude/latitude points: 72W 12N – 87.5W 20N. The location of departure (72W 12N) and the arrival location (87.5W 20N) both have a uniform uncertainty range of ±30 nm. GF2 is expected

to depart at 0000 hours on day two of the scenario with uniform departure time uncertainty of ±24 hours. The overall certainty of the intelligence about GF2 is 0.50, thus the expected detection value (q_3) of GF2 is the product of the payload and the overall certainty $q_3 = 1,500 \times 0.50 = 750$.

The fourth target (MV1) is a merchant vessel that is expected to be carrying 2,000 kg of cocaine and traveling in the Caribbean at an expected speed (U_4) of 15 knots. The expected track of MV1 is an angled route that transits through the longitude/latitude points: 75.5W 10.5N – 78W 17N – 85W 16N. The location of departure (75.5W 10.5N) and the arrival location (85W 16N) both have a uniform uncertainty range of ±40 nm. There is also a 80 nm × 80 nm uniform range of uncertainty about the waypoint (78W 17N). MV1 is expected to depart at 1200 hours on day two of the scenario with uniform departure time uncertainty of ±5 hours. The overall certainty of the intelligence about MV1 is 0.25, thus the expected detection value (q_4) of MV1 is the product of the payload and the overall certainty $q_4 = 2,000 \times 0.25 = 500$. Due to the relatively large size of a merchant vessel, the sweep width of a P-3 against this type of vessel is assumed to be 30 nm.

The fifth target (GF3) is a GO FAST boat that is expected to be carrying 1,000 kg of cocaine and traveling in the Caribbean at an expected speed (U_5) of 50 knots. The expected track of GF3 is a straight-line route that transits through the longitude/latitude points: 75.5W 10.5N – 87.5W 20N. The location of departure (75.5W 10.5N) and the arrival location (87.5W 20N) both have a uniform uncertainty range of ±20 nm. GF3 is expected to depart at 0400 hours on day three of the scenario with uniform departure time uncertainty of ±2 hours. The overall certainty of the intelligence about GF3 is 0.95, thus the expected detection value (q_5) of GF3 is the product of the payload and the overall certainty $q_5 = 1,000 \times 0.95 = 950$.

The sixth target (GF4) is a GO FAST boat that is expected to be carrying 1,000 kg of cocaine and traveling in the eastern Pacific at an expected speed (U_6) of 50 knots. The expected track of GF4 is a coastal route that transits through the longitude/latitude points: 79W 1N – 79W 5N – 83W 6N – 92.2W 14.5N. The location of departure (79W 1N) and the arrival location (92.2W 14.5N) both have a uniform uncertainty range of ±40 nm. There is also an 80 nm × 80 nm uniform range of uncertainty about each waypoint (79W 5N, 83W 6N). GF4 is expected to depart at 2300 hours on day two of the scenario with

uniform departure time uncertainty of ±2 hours. The overall certainty of the intelligence about GF1 is 0.95, thus the expected detection value (q_6) of GF6 is the product of the payload and the overall certainty $q_6 = 1,000 \times 0.95 = 950$.

The seventh target (SP2) is an SPSS that is expected to be carrying 2,500 kg of cocaine and traveling in the eastern Pacific at an expected speed (U_7) of 15 knots. The expected track of SP2 is an angled route that transits through the longitude/latitude points: 77W 5N – 86W 3N – 92.2W 14.5N. The location of departure (77W 5N) and the arrival location (92.2W 14.5N) both have a uniform uncertainty range of ±40 nm. SP2 is expected to depart at 0500 hours on day one of the scenario with uniform departure time uncertainty of ±3 hours. The overall certainty of the intelligence about SP2 is 0.50, thus the expected detection value (q_7) of SP2 is the product of the payload and the overall certainty $q_7 = 2,500 \times 0.50 = 1,250$.

The eighth target (SP3) is an SPSS that is expected to be carrying 5,000 kg of cocaine and traveling in the eastern Pacific at an expected speed (U_8) of 15 knots. The expected track of SP3 is a straight-line route that transits through the longitude/latitude points: 80W 0N – 89W 13.5N. The location of departure (80W 0N) and the arrival location (89W 13.5N) both have a uniform uncertainty range of ±35 nm. SP3 is expected to depart at 1200 hours on day one of the scenario with uniform departure time uncertainty of ±48 hours. The overall certainty of the intelligence about SP3 is 0.50, thus the expected detection value (q_8) of SP3 is the product of the payload and the overall certainty $q_8 = 5,000 \times 0.50 = 2,500$.

The ninth target (GF5) is a GO FAST boat that is expected to be carrying 2,000 kg of cocaine and traveling in the Caribbean at an expected speed (U_9) of 50 knots. The expected track of GF5 is an angled route that transits through the longitude/latitude points: 79W 1N – 92W 2S – 94W 16N. The location of departure (79W 1N) has a uniform uncertainty range of ±25 nm. The arrival location (94W 16N) has a uniform uncertainty range of ±50 nm. There is also a 50 nm × 50 nm uniform range of uncertainty about the waypoint (92W 2S). GF5 is expected to depart at 0500 hours on day one of the scenario with uniform departure time uncertainty of ±2 hours. The overall certainty of the intelligence about GF5 is 0.95, thus the expected detection value (q_9) of GF5 is the product of the payload and the overall certainty $q_9 = 2,000 \times 0.95 = 1,900$. By virtue of its large expected detection value and its relatively narrow uncertainty range, GF5 is

Figure 4.1: **Baseline scenario BS** - Spatial representation of expected target tracks in the baseline scenario. The searchers' home station is identified by "×". Nine total targets transiting the AOI over a three day planning period. Timing and uncertainty of target tracks not shown. Refer to **BS** narrative for further target details.

considered a high-value target in this scenario.

In a nine-target scenario, where target motion can be modeled as search regions moving on straight-line tracks and interdiction is not considered, developing a plan, for even a single searcher, is not an easy task for a planner. Varying target speeds and payloads, differing levels of certainty, and timing constraints over a three day period present many challenges. Clearly, since **BS** is even more complex, developing a good plan manually would be an even harder task for planners.

The remainder of this chapter successively introduces enhancements to **SSP** to build a complete model that is used to compute a combined search and interdiction plan for **BS**, while presenting intermediate numerical results for each model enhancement. In order to reduce notational complexity, each model enhancement is introduced and formulated as a stand-alone modification to **SSP**, while in practice any combination of these model enhancements can be used together to accommodate the scenario at hand. Indeed **BS**

requires them all.

4.2 Complex Target Motion

Real-world scenarios where the SSP arises can require the use of models that are more complex than those discussed thus far, e.g., **BS**. A particular target's movement track may not be along a straight line. The target may be traveling along a track that follows a particular stretch of coastline, or the target may navigate around islands or other geographic obstacles. It is also possible that the search region associated with a target changes as the target moves, perhaps due to changing weather or intelligence-driven changes to the uncertainty ranges themselves. The speed of the target may also change with ocean state conditions as a smuggler travels. All of these considerations can be modeled, at least approximately, with piecewise linear target movement tracks.

We consider the situation where target motion is nonlinear, but can be approximated by piecewise linear segments. We model the nodes N as the segmented search regions (*target segments*). Nodes represent search regions as in **SSP**, but here they do not necessary correspond to unique targets. Let $F \subseteq \hat{N}$ be the set of first segment target paths, one for each actual target. Let $B(i) \subseteq \hat{N}, i \in F$ denote the set of search region segments for each target. The piecewise linear target motion model is as follows.

Problem SSP-PWL:

$$\begin{aligned} \max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t},\boldsymbol{x},\boldsymbol{y}} \quad & \sum_{i \in F} q_i \left(1 - \exp\left\{ - \sum_{j \in B(i)} \alpha_j d_j y_j \right\} \right) \\ \text{s.t.} \quad & \text{(3.2b) – (3.2m)} \end{aligned}$$

SSP-PWL only differs from **SSP** in the objective function. In the objective we sum total search effort in the exponential. In this case the summation is over all segments in the piecewise linear target movement track. We set τ_j^{min} and τ_j^{max} in (3.2f) and (3.2g) respectively to define the connections between target path line segments.

Some computational efficiency can be realized by reducing the size of the arc set A. Clearly, any arc (i, j) where j precedes $i \in B(j)$ should be eliminated from A. This is accomplished with the set $N^{\times}(\hat{\boldsymbol{x}}_\ell)$ in constraints (3.7h). Performing this elimination

procedure for all target path line segments reduces the dimension of $\hat{\boldsymbol{X}}$ in Algorithm B&B.

4.2.1 Complex Target Motion Model Numerical Results

We now consider the baseline scenario **BS**. Model enhancements that account for multiple searchers or interdiction have not been introduced, therefore we only consider a single searcher problem with no interdiction. Additionally, we have yet to describe how to handle scenarios with multiple mission cycles. A target is considered to be *active* in a particular period if it is expected to be moving along its track at any time during the period. We restrict the scenario to first day's mission cycle (D=24 hours). Doing so removes SP1, MV1, GF3, GF4, and SP3 from the scenario because these targets are not active until after the first day. Lastly, since we have not described how to model target tracks that have a large time uncertainty window, we remove GF2 from the scenario. We denote by GF1(i) the ith segement of GF1's piecewise linear movement track, where segment 1 is the first with respect to expected departure time. We also denote by $\boldsymbol{\theta}_j^+$ and $\boldsymbol{\theta}_j^-$ the respective locations where the searcher begins and ends search for target j. In this model, $\boldsymbol{\theta}_j^+$ corresponds to the expected location of target j the moment search begins at time a_j, and $\boldsymbol{\theta}_j^-$ corresponds to the expected location of target j the moment search ends at time $a_j + d_j$.

The resulting restricted baseline scenario **BS-PWL** considers the remaining targets GF1, SP2, and GF5 as shown in Figure 4.2. During preprocessing the second segment of SP2's track can be eliminated from search consideration because the target does transit this segment until the second day of the scenario; $\tau^{min}_{\text{SP2(2)}} = 41.8$. Due to the additional segments this problem is equivalent in size to a 6-target SSP. **BS-PWL** is solved using Algorithm B&B in 3.5 seconds of computing time, yielding an expected search plan value of 1,444.1 kg of cocaine detected. The SSP heuristic correctly identifies the optimal solution in 0.6 seconds of computing time.

Table 4.1 lists the optimal search times ($\boldsymbol{a}, \boldsymbol{d}$, and $\boldsymbol{t}$) and searcher path. The rows in this table correspond to target segments and home station locations (i.e., nodes in the network G). The column labeled t lists the transit time for the searcher to arrive at each node from its previous node. The column labeled a lists the arrival time of the searcher at each node. The column labeled d lists the dwell time of the searcher at each node. The column labeled $\boldsymbol{\theta}^+$ lists the position at time of the searcher when it arrives at each

Figure 4.2: **Restricted baseline scenario BS-PWL** - Spatial representation of expected target tracks in the restricted baseline scenario **BS-PWL**. The searchers' home station is identified by "×". Timing and uncertainty of target tracks not shown. Refer to **BS** narrative for further target details.

node. The column labeled $\boldsymbol{\theta}^-$ lists the position at time of the searcher when it departs each node.

Figure 4.3 shows a spatial representation of the optimal search plan, where the dark rectangle represents the effective search region of GF5(1) in the search plan and the arrows indicate the direction of the searcher's path.

For another **SSP-PWL** example, centered on a two-searcher scenario, see Pietz and Royset (2013).

4.3 Multiple Searchers

We now consider search planning operations where a set of searchers S is available. We model this planning problem as a GOP-RDR on a searcher-expanded network $G_S = (N_S, A_S)$, where the nodes are searcher-target pairs $N_S = \{(s, j) : s \in S, j \in N\}$ and the arcs $A_S = \{(s, i, j) : s \in S, (i, j) \in A\}$ represent the transit of searcher s between

	t	a	d	$\boldsymbol{\theta}^+$	$\boldsymbol{\theta}^-$
Home	-	0	8.40	(89.1W 13.4N)	(89.1W 13.4N)
GF5(1)	2.68	11.08	4.64	(83.9W 0.1S)	(87.7W 1.0S)
Home	2.68	18.40	-	(89.1W 13.4N)	(89.1W 13.4N)

Table 4.1: **Optimal search times and searcher path for scenario BS-PWL** - The rows in this table correspond to target segments and home station locations (i.e., nodes in the network G). The column labeled t lists the transit time for the searcher to arrive at each node from its previous node. The column labeled a lists the arrival time of the searcher at each node. The column labeled d lists the dwell time of the searcher at each node. The column labeled $\boldsymbol{\theta}^+$ lists the position at time of the searcher when it arrives at each node. The column labeled $\boldsymbol{\theta}^-$ lists the position at time of the searcher when it departs each node.

Figure 4.3: **Restricted baseline scenario BS-PWL solution** - Spatial representation of **BS-PWL** optimal solution. The optimal search plan directs the searcher to search only for GF5 along its first movement track segment for a total of 4.64 hours. The resulting expected value of the search plan is 1,444.6 kg of cocaine detected. The size of the rectangular block corresponds to the total area of the search region during the time when the searcher is performing search actions in that region against GF5. Arrows indicate the direction of the searcher's path

search region i and search region j. Utilizing the vector forms of $\boldsymbol{a}_j, \boldsymbol{d}_j$ and $\boldsymbol{t}_{i,j}$ in $\mathbf{P}$, we allow each of these resource variables to have $|S|$ elements. We denote by $a_{s,j}$ and $d_{s,j}$ the respective arrival time and dwell time of searcher s in search region j, and we denote by $t_{s,i,j}$ the transit time of searcher s from search region i to search region j. The multiple

searcher SSP is stated as follows.

Problem SSP-MS:

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t},\boldsymbol{x},\boldsymbol{y}} \quad \sum_{j\in\hat{N}} q_j \left(1 - \exp\left\{-\sum_{s\in S} \alpha_{s,j} d_{s,j} y_{s,j}\right\}\right) \tag{4.2a}$$

$$\begin{aligned}
\text{s.t.} \quad & (||\boldsymbol{\rho}_{s,i} + (a_{s,i} + d_{s,i} - \tau_{s,i})\boldsymbol{u}_{s,i} - \boldsymbol{\rho}_{s,j} \\
& \ldots - (a_{s,i} + d_{s,i} + t_{s,i,j} - \tau_{s,j})\boldsymbol{u}_{s,j}|| \\
& \ldots - V t_{s,i,j}) x_{s,i,j} \leq 0, \qquad \forall (s,i,j) \in A_S \qquad (4.2b) \\
& (a_{s,i} + d_{s,i} + t_{s,i,j} - a_{s,j})\, x_{s,i,j} \leq 0, \qquad \forall (s,i,j) \in A_S \qquad (4.2c) \\
& \sum_{j\in\hat{N}_S} d_{s,j} + \sum_{(i,j)\in A_S} t_{s,i,j} \leq T_s, \qquad \forall s \in S \qquad (4.2d) \\
& \sum_{j\in N_S} d_{s,j} + \sum_{(i,j)\in A_S} t_{s,i,j} \leq D_s, \qquad \forall s \in S \qquad (4.2e) \\
& a_{s,j} \geq \tau_{s,j}^{min}, \quad \forall (s,j) \in N_S \qquad (4.2f) \\
& a_{s,j} + d_{s,j} \leq \tau_{s,j}^{max}, \quad \forall (s,j) \in N_S \qquad (4.2g) \\
& a_{s,0} = 0, \qquad \forall s \in S \qquad (4.2h) \\
& d_{s,n+1} = 0, \qquad \forall s \in S \qquad (4.2i) \\
& a_{s,j}, d_{s,j} \geq 0, \qquad \forall (s,j) \in N_S \qquad (4.2j) \\
& t_{s,i,j} \geq 0, \qquad \forall (s,i,j) \in A_S \qquad (4.2k) \\
& \boldsymbol{y}_s = \boldsymbol{\Gamma}\boldsymbol{x}_s, \quad \forall s \in S \qquad (4.2l) \\
& \boldsymbol{x}_s \in \mathbb{X}, \qquad \forall s \in S \qquad (4.2m)
\end{aligned}$$

Each expression in **SSP-MS** is a direct extension of its **SSP** counterpart where $a_{s,j}, d_{s,j}$ and $t_{s,i,j}$ are the arrival time, dwell time, and the transit time of searcher s contained in the vectors $\boldsymbol{a}_j, \boldsymbol{d}_j$ and $\boldsymbol{t}_{i,j}$ respectively. We allow resources T and D in (4.2d) and (4.2e) respectively to vary by searcher. This is a useful feature that allows the model to account for heterogeneous searchers. In the objective function (4.2a), each exponential term associated with search region j in the random search model computes detection probability by accumulating total search effort for all searchers just as in **SSP-PWL**.

4.3.1 Algorithms for the Multi-Searcher Model

B&B for the Multi-Searcher Model

Algorithm B&B can be used to solve **SSP-MS**. We modify the notation in Algorithm B&B by requiring that the nodes of the enumeration tree be viewed as (s, j)-pairs, where $s \in S, j \in N$. We also vectorize I_ℓ and $\boldsymbol{\delta}_{\ell,n+1}$ in $\mathbf{RP}(\hat{\boldsymbol{x}}_\ell)$ to account for path completion with respect to each searcher, and modify the path completion criterion in step 3 to require that $(s, j) = (s, n+1), \forall s \in S$. In principle, this can be done for an arbitrary number of searchers, however the enumeration tree grows exponentially with the number of searchers $|S|$. Fortunately, real-world applications we consider call for planning with a very limited number of searchers (i.e., one or two searchers). When searchers are homogeneous and have the same home station, many multi-searcher partial paths in the set $\hat{\boldsymbol{X}}$ are redundant because the individual partial paths for two or more searchers can be swapped to produce an equivalent multi-searcher partial path that is also in $\hat{\boldsymbol{X}}$. These redundant multi-searcher partial paths can be removed from the set $\hat{\boldsymbol{X}}$ in order to reduce the amount of enumeration that must be done in Algorithm B&B. This *redundant partial path elimination* (RPPE) belongs to the class of reduction rules used in B&B algorithms for the VRP (see Toth and Vigo 2002, ch. 2).

SSP Heuristic for the Multi-Searcher Model

In order to provide a good initial guess to Algorithm B&B, accounting for $|S|$ searchers, we define the *multi-searcher* SSP heuristic, which performs the SSP heuristic sequentially, as follows. We set the temporal clustering parameter $\Delta = 6$ hours. We initialize the path for searcher $s, s = 2, \ldots, |S|$ to each consist only of arc $(0, n+1)$. This ensures a feasible, but certainly not optimal, path for searchers 2 through $|S|$. We then run the heuristic for searcher 1 and fix the resulting path. We then do the same for searcher $s, s = 2, \ldots, |S|$ in turn. We improve the search plan by considering the removal of targets from searcher 1's path, performing Phase V of the SSP heuristic. We then do the same for searcher $s, s = 2, \ldots, |S|$ in turn. Lastly, we attempt to improve the plan by allowing each pair searchers to swap their entire search paths. The modified heuristic returns the best search plan encountered after the aforementioned steps are completed. We do not consider pairwise target swaps between searchers nor do we consider a parallel implementation of the SSP heuristic. Our aim is to quickly provide a good initial solution to Algorithm B&B. To make notation more concise, we use the term SSP heuristic more broadly to refer either the (single-searcher) SSP heuristic or the multi-searcher SSP heuristic as the

problem context dictates. The long-form name, multi-searcher SSP heuristic, is used when the distinction between the single-searcher algorithm and the multi-searcher algorithm is important.

Cyclic Seesaw Heuristic for the Multi-Searcher Model

We propose another heuristic, the Cyclic Seesaw (CS) heuristic, for solving multiple searcher SSPs. This heuristic is inspired by a decomposition approach to optimizing over vector valued variables (Spall 2012). The fundamental idea of this cyclic seesaw method is to fix a subset of the decision variables in **SSP-MS** and solve for the remainder, alternating or cycling between fixing and solving until a stable point is reached. In **SSP-MS** a natural place to partition decision variables $\boldsymbol{x}$ is by searcher. It should be noted that since the decision variables $\boldsymbol{x}$ are integer valued, this approach is not guaranteed to converge to the optimal solution (Spall 2012). We now describe how the CS heuristic is implemented.

CS Heuristic: We begin by defining the cycle limit parameter $\Lambda \geq 1$ and initialization parameter $\iota \in \{0, 1\}$. We initialize the cycle count $\lambda = 0$.

Phase I. We initialize the search plan as follows. If $\iota = 0$, we initialize the search plan $\boldsymbol{x}$ for all searchers to the *null path*, where $x_{s,0,n+1} = 1, \forall s \in S$ and $x_{s,i,j} = 0, \forall s \in S, \forall (i,j) \in A \backslash \{(0, n+1)\}$. If $\iota = 1$, we initialize the search plan $\boldsymbol{x}$ for all searchers using the SSP heuristic. We save the initial search plan as $\bar{\boldsymbol{x}}$.

Phase II. We do the following for each searcher s in turn. We fix the plan for all searchers other than s to $\boldsymbol{x_{s'}} = \bar{\boldsymbol{x}}_{\boldsymbol{s'}}, \forall s' \in S \backslash \{s\}$. Using Algorithm B&B we solve for the optimal search plan with respect to searcher s. We save the resulting solution as the new incumbent $\bar{\boldsymbol{x}}_{\boldsymbol{s}}$.

Phase III. We increment cycle count λ, then repeat Phase II until $\lambda = \Lambda$. Upon termination return the last incumbent search plan $\bar{\boldsymbol{x}}$.

We observe that the multi-searcher SSP heuristic can be viewed as using the (single-searcher) SSP heuristic in place of Algorithm B&B within the CS heuristic.

4.3.2 Multi-Searcher Model Numerical Results

We now return to the restricted baseline scenario **BS-PWL**. Including the second searcher we arrive at the still restricted, but modified, scenario **BS-MS**. We observe that since the searchers are homogeneous and have the same home station, the respective search plans for searcher 1 and searcher 2 can always be swapped to produce an equivalently valued search plan. We take advantage of this fact using RPPE in Algorithm B&B. The SSP heuristic produces a search plan that has an objective function value of 2,035.1 kg of cocaine detected in 1.3 seconds of computing time. The search times and searcher path for this solution are shown in Table 4.2.

Searcher	Segment	t	a	d	$\boldsymbol{\theta}^+$	$\boldsymbol{\theta}^-$
1	Home	-	0	2.54	(89.1W 13.4N)	(89.1W 13.4N)
	GF1(2)	1.83	4.37	2.68	(79.8W 10.0N)	(82.0W 10.0N)
	GF5(1)	1.79	8.85	0.98	(82.1W 0.3N)	(82.9W 0.1N)
	Home	2.71	12.54	-	(89.1W 13.4N)	(89.1W 13.4N)
2	Home	-	0	8.40	(89.1W 13.4N)	(89.1W 13.4N)
	GF5(1)	2.68	11.08	4.64	(83.9W 0.1S)	(87.7W 1.0S)
	Home	2.68	18.40	-	(89.1W 13.4N)	(89.1W 13.4N)

Table 4.2: **SSP heuristic search times and searcher path for scenario BS-MS** - Expected value of the SSP heuristic search plan is 2,035.1 kg of cocaine detected.

The optimal search plan, computed using Algorithm B&B with RPPE in 48.9 seconds, has an objective function value of 2,254.6 kg of cocaine detected. Since real-world scenarios do not in general have homogeneous searchers with the same home station, it is not always appropriate to use RPPE. The optimal solution is computed using Algorithm B&B without RPPE in 2.7 minutes. Clearly RPPE yields a substantial runtime savings, 70% in this case, and should be used when appropriate. For the remainder of this chapter, Algorithm B&B applied to **BS** uses RPPE and runtimes reflect this more efficient approach.

The search times and searcher path for the optimal solution are shown in Table 4.3. Figure 4.4 shows a spatial representation of the optimal search plan.

The SSP heuristic solution differs from the optimal solution in that SP2 is removed from the heuristic ordering due to a relatively low rate-of-reward $\hat{\beta}$, which is mainly driven by the small sweep width of a P-3 against an SPSS (see Table 4.4). Timing and travel distance, however, make SP2 a worthwhile target to include in the optimal search plan.

Searcher	Segment	t	a	d	$\boldsymbol{\theta}^+$	$\boldsymbol{\theta}^-$
1	Home	-	0	8.40	(89.1W 13.4N)	(89.1W 13.4N)
	GF5(1)	2.68	11.08	4.64	(83.9W 0.1S)	(87.7W 1.0S)
	Home	2.68	18.40	-	(89.1W 13.4N)	(89.1W 13.4N)
2	Home	-	0	6.39	(89.1W 13.4N)	(89.1W 13.4N)
	GF1(3)	1.37	7.76	1.91	(82.2W 10.6N)	(82.8W 12.0N)
	SP2(1)	1.57	11.24	2.69	(78.5W 4.7N)	(79.2W 4.5N)
	Home	2.46	16.39	-	(89.1W 13.4N)	(89.1W 13.4N)

Table 4.3: **Optimal search times and searcher path for scenario BS-MS** - Expected value of the SSP heuristic search plan is 2,254.6 kg of cocaine detected.

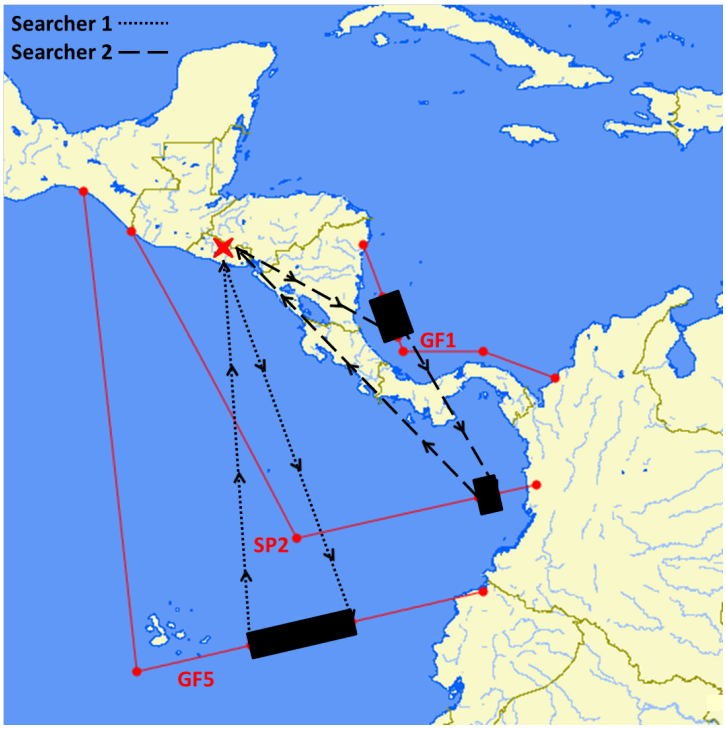


Figure 4.4: **Restricted baseline scenario BS-MS solution** - Spatial representation of **BS-MS** optimal solution. The resulting expected value of the search plan is 2,254.6 kg of cocaine detected. The size of the rectangular blocks correspond to the total area of the search region during the time when the searcher is performing search actions.

The CS heuristic, with $\Lambda = 1$ and $\iota = 1$, correctly identifies the optimal search plan in 8.3 seconds of computing time. Clearly this heuristic will not always produce the optimal solution, but in this case it does so because in Phase I the SSP heuristic provides an initial plan which is optimal with respect to one of the searchers.

We refer to Pietz and Royset (2013) for an **SSP-MS** example which studies the differences

Segment	$\hat{\beta}$	q	α	W	$\tilde{\tau}\tilde{\rho}$
GF1(1)	214.7	950	0.256	15	12000
GF1(2)	214.7	950	0.256	15	12000
GF1(3)	214.7	950	0.256	15	12000
SP2(1)	187.7	1250	0.163	5	6300
GF5(1)	503.0	1900	0.308	15	10000
GF5(2)	332.2	1900	0.192	15	16000

Table 4.4: **Rate-of-reward parameter comparison of scenario BS-MS target segments** - The small sweep width associated with SP2 make it an unfavorable target in the SSP heuristic search order.

between multiple-searcher plans done manually, plans obtained with the SSP heuristic, and plans computed using Algorithm B&B.

4.4 Fixed Region Search

Suppose that the uncertainty associated with a target track is such that defining a moving search region is not realistic. This can happen, for example, when the departure time uncertainty range is large or when there is unpredictable variation in the target speed. In this case, we model the search for targets using a random search model (Washburn 2002, ch. 2) applied to a static *fixed region*. In keeping with the SSP, we assume that the probability distribution of the target within the search region is uniform.

In a sense this model is less complex than that of the SSP which considers moving search regions. However, a subtle practical matter must be considered. In the SSP, we assume that the moving regions are not too large. As such, the question of where within the region a searcher is routed to and from is insignificant. Out of convenience we choose the center of the region. When the search region is small, the chosen tactical search pattern can be accommodated with a negligible difference in the searcher's entry and exit locations in this region. In the situation when the search region is large, we must account for the difference in entry and exit points. We do this by including the entry and exit locations in the model as decision variables.

We model searcher transit to and from a convex subset $\mathcal{R}_j$ of the fixed region associated with target j. We denote by $\boldsymbol{r}_j^+$ and $\boldsymbol{r}_j^-$ the respective entry and exit location of the searcher in fixed region j. We model both moving and fixed regions by introducing the respective disjoint subsets Q and R, where $Q \cup R = N$ are the nodes in the directed network G. We include the home station and recovery location in the set Q. It is

convenient to define $\hat{Q} = Q \backslash \{0, n+1\}$. Directed arcs are defined between all $(i,j) \in A$ as follows. Let A_{QQ} be the set of arcs connecting a moving search region to another moving search region. Let A_{QR} be the set of arcs connecting a moving search region to fixed search region. A_{QR} and A_{RR} are defined similarly. We introduce the vectorized location notation $\boldsymbol{r}_j = (\boldsymbol{r}_j^+, \boldsymbol{r}_j^-)$, and $\boldsymbol{r} = (\boldsymbol{r}_{j_1}, \boldsymbol{r}_{j_2}, \ldots, \boldsymbol{r}_{j_{|R|}})$. Accounting for fixed region targets, the modified SSP is stated as follows.

Problem SSP-FR:

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t},\boldsymbol{r},\boldsymbol{x},\boldsymbol{y}} \quad \sum_{j \in \hat{N}} q_j \left(1 - \exp\left\{-\alpha_j d_j y_j\right\}\right) \tag{4.3a}$$

$$\text{s.t.} \quad (||\boldsymbol{\rho}_i + (a_i + d_i - \tau_i)\boldsymbol{u}_i - \boldsymbol{\rho}_j \ldots - (a_i + d_i + t_{i,j} - \tau_j)\boldsymbol{u}_j|| \ldots - V t_{i,j}) x_{i,j} \leq 0, \quad \forall (i,j) \in A_{QQ} \tag{4.3b}$$

$$(||\boldsymbol{\rho}_i + (a_i + d_i - \tau_i)\boldsymbol{u}_i - \boldsymbol{r}_j^+|| \ldots - V t_{i,j}) x_{i,j} \leq 0, \quad \forall (i,j) \in A_{QR} \tag{4.3c}$$

$$(||\boldsymbol{r}_i^- - \boldsymbol{\rho}_j - (a_i + d_i + t_{i,j} - \tau_j)\boldsymbol{u}_j|| \ldots - V t_{i,j}) x_{i,j} \leq 0, \quad \forall (i,j) \in A_{RQ} \tag{4.3d}$$

$$(||\boldsymbol{r}_i^- - \boldsymbol{r}_j^+|| - V t_{i,j}) x_{i,j} \leq 0, \quad \forall (i,j) \in A_{RR} \tag{4.3e}$$

$$(||\boldsymbol{r}_j^- - \boldsymbol{r}_j^+|| - \hat{V} d_j) y_j \leq 0, \quad \forall j \in R \tag{4.3f}$$

$$\boldsymbol{r}_j^-, \boldsymbol{r}_j^+ \in \mathcal{R}_j, \quad \forall j \in R \tag{4.3g}$$

$$(3.2\text{c}) - (3.2\text{m})$$

Constraints (4.3b) through (4.3e) ensure that the time it takes the searcher to travel between each pair of regions, moving or fixed, is feasible given its maximum cruise speed V. Constraint (4.3f) ensures that the time it takes the searcher to travel from the entry point to the exit point of each fixed region is feasible given its on-station speed $\hat{V}$. Constraint (4.3g) restricts the searcher's entry point and exit point in each fixed region j to lie within the convex feasibility set $\mathcal{R}_j$. We observe that (4.3g) need not depend on $\boldsymbol{x}$ because $\boldsymbol{r}_j$ is a vacuous variable when region j is not searched.

4.4.1 Fixed Region Search Model Numerical Results

Returning to **BS-MS**, we also include for day 1 the target that exhibit large departure time uncertainty ranges, namely GF2. We assume that the fixed region entry and exit points are constrained to $\mathcal{R}_{\text{GF2}}$ the line segment between the expected departure and arrival locations of GF2; 72W 12N - 87.5W 20N. We arrive at the modified scenario **BS-FR** with expected target tracks shown in Figure 4.5.

Figure 4.5: **Restricted baseline scenario BS-FR** - Spatial representation of expected target tracks in the restricted baseline scenario **BS-FR**. The searchers' home station is identified by "×". Timing and uncertainty of target tracks not shown. Refer to **BS** narrative for further target details.

This scenario **BS-FR** is solved using the SSP heuristic and Algorithm B&B without any further modifications. The SSP heuristic returns the same search plan reported in Section 4.3.2 (Table 4.2) in 1.4 seconds of computing time. The optimal solution, computed in 64.8 seconds, is also the same as reported in Section 4.3.2 (Table 4.3 and Figure 4.4). The large search area associated with GF2 results in a search value ($\hat{\beta} = 15.8$) that is an order of magnitude smaller than that of the other targets (see Table 4.4). As a result, it is not beneficial for either searcher to consume transit time in order to search for GF2.

To illustrate a search plan that includes a fixed region search in the optimal solution we consider an excursion to **BS-FR**, adjusting the intelligence data for GF2 as follows. The expected departure time is shifted 12 hours earlier to 1200 on day 1. The expected payload of GF is increased to 5,000 kg of cocaine. Additionally, the overall certainty of the intelligence is increased to 0.95. The later two modifications serve to increase the search value of GF2, while the first modification provides a greater opportunity for seach by shifting τ^{min}_{GF2} to earlier in day 1. We denote by **BS-FR-E** this excursion to **BS-FR** that includes these three modifications.

The optimal solution for **BS-FR-E**, computed in 2.3 minutes, yields an expected search value of 2,297.0 kg of cocaine detected. The optimal search plan is given in Table 4.5. Note that $\boldsymbol{\theta}^+_j = \boldsymbol{r}^+_j$ and $\boldsymbol{\theta}^-_j = \boldsymbol{r}^-_j$ for $j \in R$, while $\boldsymbol{\theta}^+_j$ and $\boldsymbol{\theta}^-_j$ for $j \in Q$ represent expected locations as described in Section 4.2.1.

Searcher	Segment	t	a	d	$\boldsymbol{\theta}^+$	$\boldsymbol{\theta}^-$
	Home	-	0	8.40	(89.1W 13.4N)	(89.1W 13.4N)
1	GF5(1)	2.68	11.08	4.64	(83.9W 0.1S)	(87.7W 1.0S)
	Home	2.68	18.40	-	(89.1W 13.4N)	(89.1W 13.4N)
	Home	-	0	5.54	(89.1W 13.4N)	(89.1W 13.4N)
	GF1(2)	1.46	7.00	0.06	(81.9W 10.0N)	(82.0W 10.0N)
2	GF1(3)	0.00	7.06	3.13	(82.0W 10.0N)	(82.9W 12.4N)
	GF2	0.85	11.03	3.30	(80.8W 16.5N)	(86.1W 19.3N)
	Home	1.21	15.54	-	(89.1W 13.4N)	(89.1W 13.4N)

Table 4.5: **Restricted baseline scenario excursion BS-FR-E solution** - The plan for searcher 1 is unchanged relative to the **BS-MS** optimal solution, searcher 2 is now directed to GF2 instead of SP2.

Figure 4.6 depicts the optimal search plan for **BS-FR-E**. We observe that the optimal plan for searcher 1 is unchanged with respect to the optimal plan for **BS-FR**, searching for GF5 with the same search times. The optimal plan for searcher 2, however, shifts to search GF1 earlier in day 1 and then proceeds to search for GF2 in its fixed region. Recall that search is assumed to be carried out according the random search model, and the planner chooses the tactical pattern to be flown by the searchers in each search region. The large shaded search area for GF2 shown in Figure 4.6 reflects the fixed geographical boundaries of the search region, and is not meant to imply that search is required be conducted everywhere in this region. In fact, searching this entire fixed region is not possible given the 3.3 hour search dwell time. The GF2 search region is over 1000 nm long ($|\bar{\boldsymbol{\rho}}_{\text{GF2}} - \boldsymbol{\rho}_{\text{GF2}}| = 1014$ nm). Searcher 2 traveling at $\hat{V} = 205$ knots would take almost

five hours just to transit across the region. The optimal entry and exit locations $\boldsymbol{r}^{+}_{\mathrm{GF2}}$ and $\boldsymbol{r}^{-}_{\mathrm{GF2}}$ are chosen to reduce transit time as much as possible provided that constraints (4.3f) and (4.3g) are satisfied.

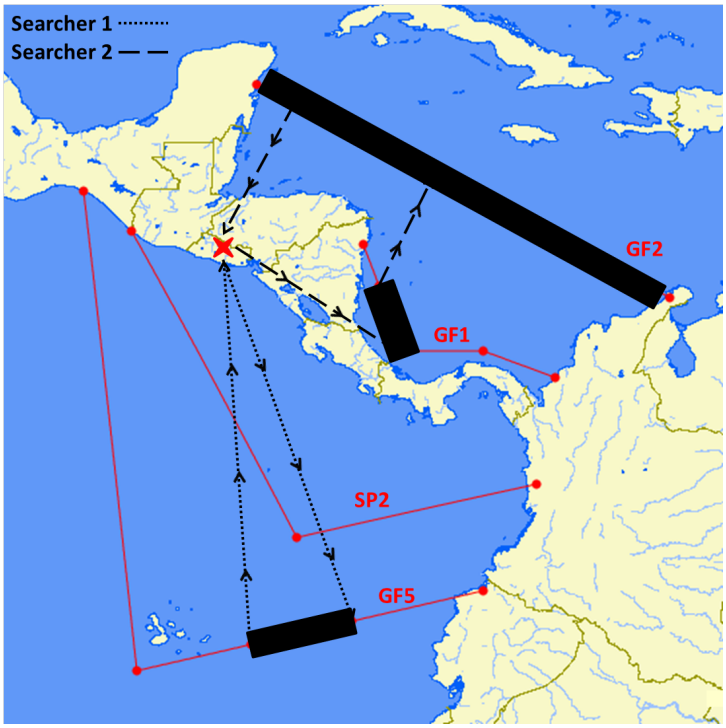


Figure 4.6: **Restricted baseline scenario excursion BS-FR-E solution** - Searcher 1 is routed to search GF5, while searcher 2 is routed to search GF1 and GF2. The path of searcher 2 into and out of the fixed search region for GF2 reflects the optimal values for $\boldsymbol{r}^{-}$ and $\boldsymbol{r}^{+}$ respectively. Within the search region (shaded dark) search actions are carried out according to the random search model.

4.5 Interdiction

We now consider search planning operations where the searcher is routed in the AOI to detect moving targets, and when a detection occurs, an interdictor is called to respond. While the interdictor is in transit to capture the detected target, the searcher monitors the target to prevent escape. At the moment the target is detected, one of three things can happen. First, the target could take evasive action by altering its intended track in an attempt to evade capture. This possibility is not considered in our one-sided search model. Search games are more appropriate models to deal with this consideration (see, e.g., Alpern and Gal 2003). Second, the target could continue on its intended track. Third, the target could stop. The latter two possibilities can be captured by our search

model by expressing the distance between the target and the interdictor using a second order cone constraint. We proceed under the assumption that targets stop when detected by a searcher.

Since the endurance of the searcher and the speed of the interdictor are both limited, we must account for the distance between the position of the interdictor and the location where a detection may occur. The interdictor may be positioned at a fixed standby location that cannot be changed by the search controller. In this case, the feasible search area is constrained by the interdictor's location. Alternatively, the standby location of the interdictor may be a decision variable. Yet another possibility exists where the search controller may specify an initial standby position for the interdictor, as well as a constant drift course and speed (*drift vector*). We model the later as it is the more general case.

Let $\boldsymbol{p} \in \mathcal{P}$ be the initial standby position of the surface interdictor, where $\mathcal{P}$ is a convex set. Let $\hat{\boldsymbol{p}} \in \mathcal{P}$ be the final resting position of the surface interdictor. Let $\boldsymbol{v}$ be the drift vector of the surface interdictor. Let V_I be the intercept speed of the interdictor. We assume that if a detection occurs in any searched region, the searcher will stay on station as required so that the interdictor may respond. Therefore, the interdictor must be able to travel to the search region before the planning period expires and before the searcher reaches its endurance limit. In the unlikely event that multiple detections occur at the same time for the same interdictor, we assume that the interdictor will respond to the first call it receives from a searcher. We also assume that the searcher maintains a safety stock of fuel so that, when it has to monitor a detected target, it is still able to return to home station at the end of its mission. We account for safety stock through *backtracking time* b_j, which is the minimum time required for the searcher to return to home station from any point on target segment j that is assigned to be searched. Backtracking time can be modeled as a fixed value, possibly different for each target, or as a decision variable. We choose the latter as it is the more general case. The interdictor coordinated SSP is stated as follows.

Problem SSP-I:

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t},\boldsymbol{b},\boldsymbol{p},\hat{\boldsymbol{p}},\boldsymbol{x},\boldsymbol{y}} \quad \sum_{j\in N} q_j \left(1-\exp\left\{-\alpha_j d_j y_j\right\}\right) \tag{4.4a}$$

$$\text{s.t.} \quad (||\boldsymbol{\rho}_i + (a_i-\tau_i)\boldsymbol{u}_i - \boldsymbol{p} - a_i\boldsymbol{v}|| \\ \ldots - V_I(T+d_0-b_i-a_i))y_i \le 0, \quad \forall i \in \hat{N} \tag{4.4b}$$

$$(||\boldsymbol{\rho}_i + (a_i+d_i-\tau_i)\boldsymbol{u}_i - \boldsymbol{p} - (a_i+d_i)\boldsymbol{v}|| \\ \ldots - V_I(T+d_0-b_i-a_i-d_i))y_i \le 0, \quad \forall i \in \hat{N} \tag{4.4c}$$

$$(||\boldsymbol{\rho}_i + (a_i-\tau_i)\boldsymbol{u}_i - \boldsymbol{p} - a_i\boldsymbol{v}|| \\ \ldots - V_I(D-b_i-a_i))y_i \le 0, \quad \forall i \in \hat{N} \tag{4.4d}$$

$$(||\boldsymbol{\rho}_i + (a_i+d_i-\tau_i)\boldsymbol{u}_i - \boldsymbol{p} - (a_i+d_i)\boldsymbol{v}|| \\ \ldots - V_I(D-b_i-a_i-d_i))y_i \le 0, \quad \forall i \in \hat{N} \tag{4.4e}$$

$$(||\boldsymbol{\rho}_i + (a_i-\tau_i)\boldsymbol{u}_i - \boldsymbol{\rho}_{n+1}|| - Vb_i)y_i \le 0, \quad \forall i \in \hat{N} \tag{4.4f}$$

$$(||\boldsymbol{\rho}_i + (a_i+d_i-\tau_i)\boldsymbol{u}_i - \boldsymbol{\rho}_{n+1}|| - Vb_i)y_i \le 0, \quad \forall i \in \hat{N} \tag{4.4g}$$

$$\boldsymbol{p} + a_{n+1}\boldsymbol{v} - \hat{\boldsymbol{p}} = \boldsymbol{0} \tag{4.4h}$$

$$\boldsymbol{p}, \hat{\boldsymbol{p}} \in \mathcal{P} \tag{4.4i}$$

(3.2b) – (3.2m)

Constraints (4.4b) and (4.4c) require that the distance between where search occurs and position of interdictor is within the travel range of the searcher given the amount of time remaining relative to the endurance limit of the searcher. Figures 4.7 and 4.8 illustrate how constraints (4.4b) and (4.4c) function in this model. We assume that the maximum speed of the interdictor is faster than its intercept speed. This allows the interdictor to successfully respond when the searcher detects a target within its search region and just outside of the interdictor response circle. Similarly, constraints (4.4d) and (4.4e) require that the distance between where search occurs and position of interdictor is within the travel range of the searcher given the amount of time remaining in the planning cycle. Constraints (4.4f) and (4.4g) require that the distance between where search occurs and searcher's home station is within the backtracking time travel range of the searcher. Constraint (4.4h) sets the final resting position of the interdictor to be its drifted position at the moment the searcher returns to home station. Constraint (4.4i) requires that the

position of the interdictor remain within the feasibility set $\mathcal{P}$ throughout the mission.

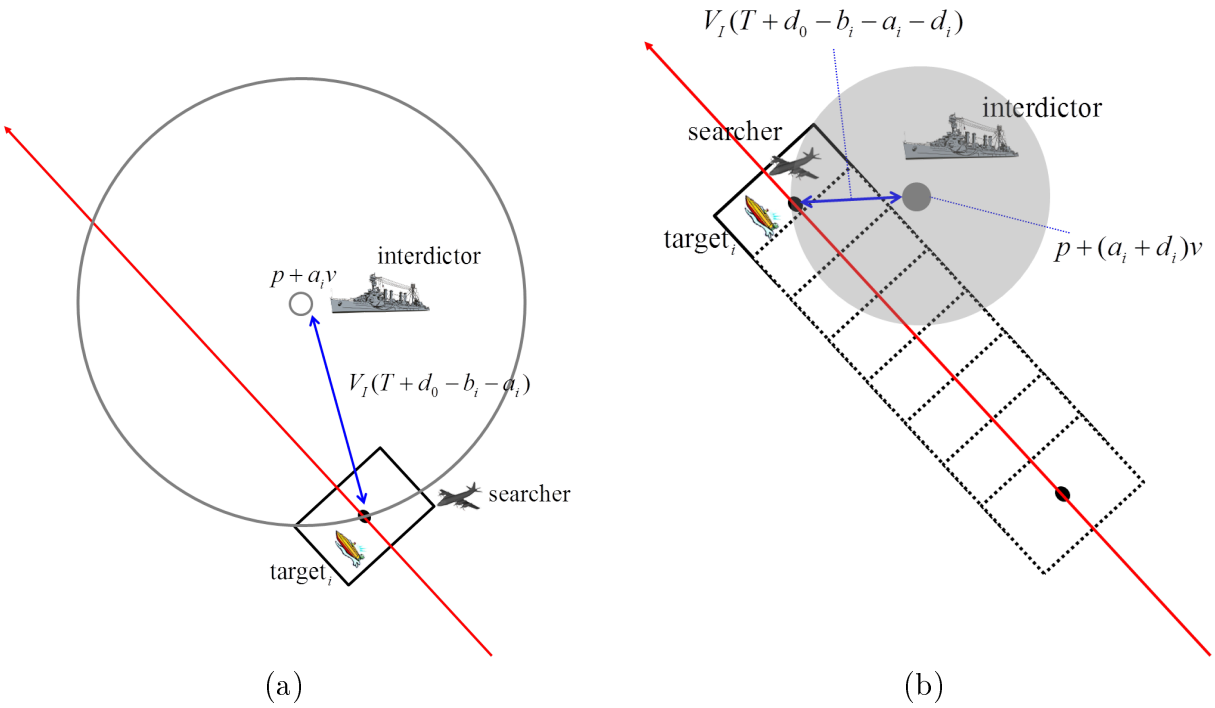


Figure 4.7: **Interdiction model diagram -** Consider a straight line movement track for target i and rectangular search region as shown. (a) Constraint (4.4b) requires that the distance between where search begins against target i and the position $\boldsymbol{p}+a_i\boldsymbol{v}$ of the interdictor is no greater than the range of the interdictor $V_I(T+d_0-b_i-a_i)$. The search region at the moment search begins against target i is shown as the solid rectangle. The interdictor response circle with radius $V_I(T+d_0-b_i-a_i)$ and center $\boldsymbol{p}+a_i\boldsymbol{v}$ is shown. (b) Constraint (4.4c) requires that the distance between where search ends against target i and the drifted position $\boldsymbol{p}+(a_i+d_i)\boldsymbol{v}$ of the interdictor no greater than the range of the interdictor $V_I(T+d_0-b_i-a_i-d_i)$. The search region at the moment search ends against target i is shown as the solid rectangle. The search region moving over time epochs is shown as indicated by the dotted rectangles. The interdictor response circle with radius $V_I(T+d_0-b_i-a_i-d_i)$ and center $\boldsymbol{p}+(a_i+d_i)v$ are shown.

The special case where the position of the interdictor is fixed and unchangeable can be modeled by fixing $\boldsymbol{p}$ and $\boldsymbol{v}=\boldsymbol{0}$. Similarly, other situations can be modeled by allowing $\boldsymbol{p}$ to vary and/or choosing $\boldsymbol{v}\neq\boldsymbol{0}$. Multiple interdictors are easily handeled with this model by indexing the variables and parameters associated with the interdictor. If each target can be matched to an interdictor, the multiple interdictor model is no more difficult

to solve than the single interdictor model. When the target-to-interdictor matching is unknown or cannot reasonably be set manually, this introduces integer variables that must be handled along with $\boldsymbol{x}$ through branching. Since predetermined patrol zones and state sovereignty often drive surface interdictor placement, we assume that the target-to-interdictor matching can be done manually.

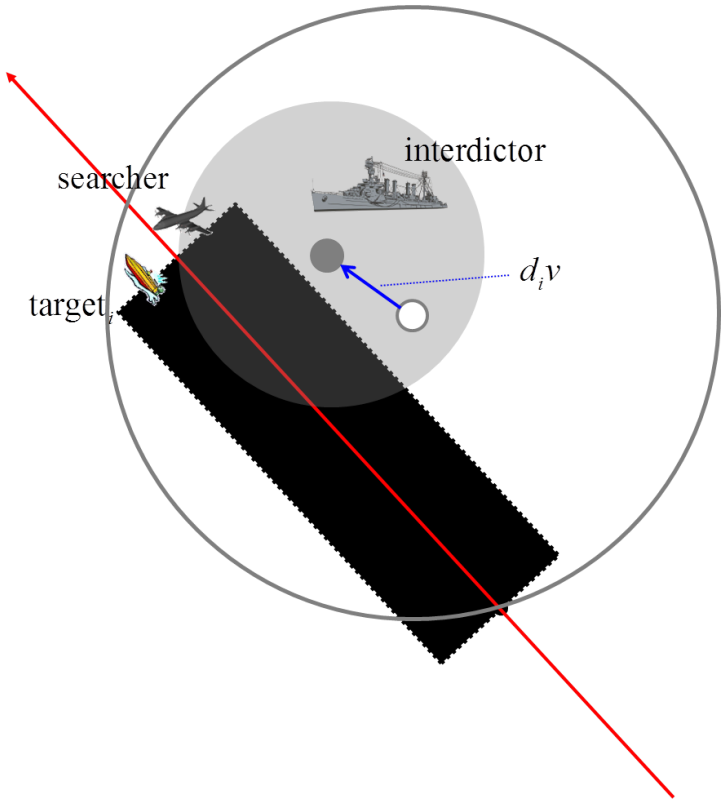


Figure 4.8: **Interdiction model diagram combined** - We combine the both parts of Figure 4.7 by shading the response circle associated with constraint (4.4c) and filling in the moving search region as done in previous illustrations. Interdiction model solution figures later in this section will have this representation. We see that the center of the shaded response circle is offset by the vector $d_i\boldsymbol{v}$. We observe that when $\boldsymbol{v} = \mathbf{0}$, the centers of the response circles must be the same position $\boldsymbol{p}$.

4.5.1 Interdiction Model Numerical Results

Returning to **BS** we assume that interdictors are assigned in regional zones as indicated in Table 4.6. Recall that the intercept speed of the interdictor is $V_I = 20$ knots.

We augment the restricted scenario **BS-FR** by adding the four available interdiction assets. We assume that backtracking time for each searcher is included in its endurance time T; we fix $b_j = 0, \forall j \in N$, and we relax constraints (4.4f) and (4.4g). In keeping with

Interdictor	Targets	Zone
1	GF2, GF3, MV1	NE Caribbean zone
2	GF1, SP1	SW Caribbean zone
3	GF4, SP2, SP3	NE Eastern Pacific zone
4	GF5	SW Eastern Pacific zone

Table 4.6: **Zonal interdictor-to-target assignment for scenario BS**

BS in Section 4.1, we do not allow the interdictors to change position after their location has been coordinated with the search plan. Therefore, we require $\boldsymbol{v} = \mathbf{0}$ and $\boldsymbol{p} = \hat{\boldsymbol{p}}$. We restrict the positions of interdictors 1 and 2 to be within the Caribbean Sea, and the positions of interdictors 3 and 4 to be within the eastern Pacific Ocean. Accordingly, feasible polyhedral sets $\mathcal{P}_k$ for interdictors $k = 1, 2, 3, 4$ are defined by the convex hulls of the sets of extreme points given in Table 4.7. We denote by **BS-I** this restricted scenario that includes interdictior coordination.

Interdictor (k)	Extreme Points of $\mathcal{P}_k$
1	(70W 25N), (90W 22N), (82W 10N), (76.3W 9N), (72W 12N)
2	(70W 25N), (90W 22N), (82W 10N), (76.3W 9N), (72W 12N)
3	(100W 5S), (81W 5S), (77W 6N), (100W 17N)
4	(100W 5S), (81W 5S), (77W 6N), (100W 17N)

Table 4.7: **Interdictor position polyhedral sets extreme points**

Scenario **BS-I** is solved using the SSP heuristic and Algorithm B&B. The SSP heuristic returns the search plan and interdictor placements reported in Table 4.8 and Table 4.9 respectively in 1.5 seconds of computing time. Expected value of the SSP heuristic search plan is 1,978.4 kg of cocaine detected. While the searcher path $\boldsymbol{x}$ in this plan is equivalent to the SSP heuristic searcher path for **BS-FR** reported in Section 4.3.2, the search times and objective function value are diminished due to the restricted range of the interdictors. We observe that interdictors 1 and 3 are not used in the SSP heuristic plan because GF2 and SP2 are not searched.

The optimal search plan and interdictor placements, computed in 84.8 seconds, are reported in Table 4.10 and Table 4.11. Expected value of the optimal search plan is 2,189.8 kg cocaine detected. The optimal search plan differs from the optimal search plan for **BS-FR** reported in Section 4.3.2 in how the search effort for GF5 is allocated. In **BS-I**, it is optimal to search GF5 along both of its movement track segments. As with the heuristic solution, this is due to the restricted range of the interdictor.

Searcher	Segment	t	a	d	$\boldsymbol{\theta}^+$	$\boldsymbol{\theta}^-$
1	Home	-	0	2.86	(89.1W 13.4N)	(89.1W 13.4N)
	GF1(2)	1.79	4.65	2.41	(78.0W 10.0N)	(82.0W 10.0N)
	GF5(1)	1.79	8.85	1.31	(82.1W 0.3N)	(83.2W 0.0N)
	Home	2.70	12.86	-	(89.1W 13.4N)	(89.1W 13.4N)
2	Home	-	0	4.77	(89.1W 13.4N)	(89.1W 13.4N)
	GF5(1)	2.80	7.57	4.11	(81.1W 0.5N)	(84.4W 0.3S)
	Home	2.67	14.36	-	(89.1W 13.4N)	(89.1W 13.4N)

Table 4.8: **SSP heuristic search times and searcher path for scenario BS-I** - Expected value of the SSP heuristic search plan is 1,978.4 kg of cocaine detected.

Interdictor	Position
1	-
2	(81.9W 10N)
3	-
4	(83.4W 0.0S)

Table 4.9: **SSP heuristic interdictor standby positions for scenario BS-I** - Interdictors 1 and 3 are not used in the SSP heuristic plan because GF2 and SP2 are not searched.

Searcher	Segment	t	a	d	$\boldsymbol{\theta}^+$	$\boldsymbol{\theta}^-$
1	Home	-	0	14.16	(89.1W 13.4N)	(89.1W 13.4N)
	GF5(1)	2.71	16.87	4.14	(88.6W 1.2S)	(92.0W 2.0S)
	GF5(2)	0.00	21.01	0.10	(92.0W 2.0S)	(92.0W 1.9S)
	Home	2.89	24.00	-	(89.1W 13.4N)	(89.1W 13.4N)
2	Home	-	0	6.39	(89.1W 13.4N)	(89.1W 13.4N)
	GF1(3)	1.37	7.76	1.91	(82.2W 10.6N)	(82.8W 12.0N)
	SP2(1)	1.57	11.24	2.69	(78.5W 4.7N)	(79.2W 4.5N)
	Home	2.46	16.39	-	(89.1W 13.4N)	(89.1W 13.4N)

Table 4.10: **Optimal search times and searcher path for scenario BS-I** - Expected value of the optimal search plan is 2,189.8 kg cocaine detected.

Figure 4.9 shows a spatial representation of the optimal search plan. As the interdictor response circles indicate, the search plan relative to GF1 and SP2 are not limited by the range of the interdictor. The GF1-interdictor 2 range circles could be shifted in the northwest direction and the search plan would still be feasible. Similarly, the SP2-interdictor 3 range circles could be shifted east. This is not the case for the GF5-interdictor 4 range circles. The edge of the outer interdictor range circle is tanget to the point $\boldsymbol{\theta}^+_{\text{GF5(1)}}$ where search begins against GF5, while the edge of the inner interdictor range circle is tanget to the point $\boldsymbol{\theta}^-_{\text{GF5(2)}}$ where search ends against GF5. This indicates that the search plan relative to GF5 is limited by the range of the interdictor, and it highlights why the

Interdictor	Position
1	-
2	(81.9W 10N)
3	(79.9W 4.1N)
4	(91.0W 1.7S)

Table 4.11: **Optimal interdictor standby positions for scenario BS-I** - Interdictor 1 is not used in the optimal plan because GF2 is not searched.

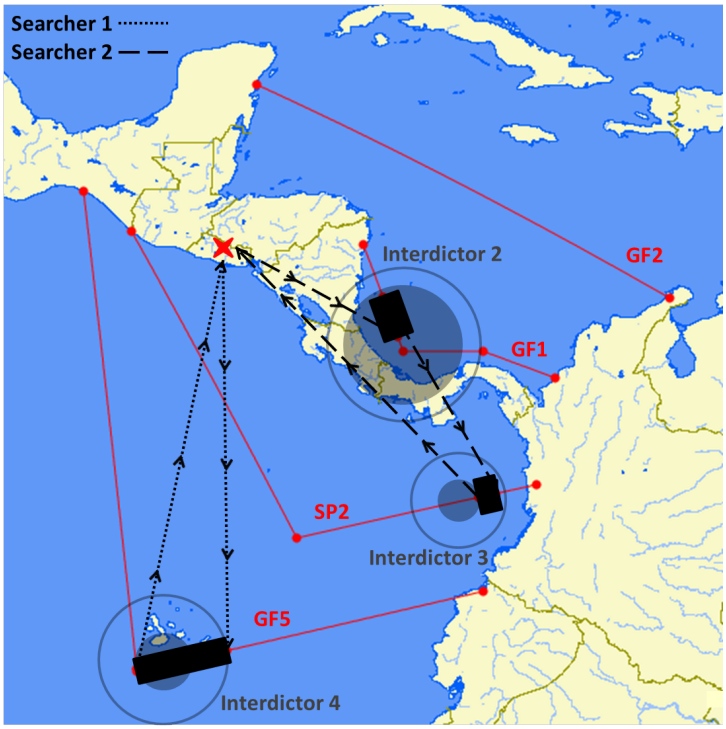


Figure 4.9: **Restricted baseline scenario BS-I solution** - Spatial representation of **BS-I** optimal solution. The resulting expected value of the search plan is 2,189.8 kg of cocaine detected, which is less that the expected search plan value associated with **BS-FR** due to the restrictions induced by the interdictor ranges. The size of the rectangular blocks correspond to the total area of the search region during the time when the searcher is performing search actions. The large transparent outlined outer circles, with optimal interdictor standby locations at the center, represent the response range, based on constraints (4.4b), of each respective interdictor at the moment search begins for its assigned target. The small shaded inner circles, again with optimal interdictor standby locations at the center, represent the response range, based on constraints (4.4c), of each respective interdictor at the moment search ends for its assigned target.

optimal search plan for **BS-I** differs from that of **BS-FR**.

One might consider using the optimal solution for **BS-FR** and simply adding in the interdictor position later by solving a single NLP with decision variable vector $\boldsymbol{p}$ (e.g., solve **SSP-I** by fixing all variables except $\boldsymbol{p}$). Such as strategy may appear practical

in situations where interdictor coordination is an afterthought, an optimal solution to **SSP-FR** is already in hand, and there is not enough time to set-up and solve **SSP-I**. Unfortunately this approach does not work in general, because there is no guarantee that the incumbent search plan is feasible with respect to the interdictor. Taking the optimal search plan solution to **BS-FR** and overlaying the interdictor response circles we arrive at Figure 4.10. Given 4.64 hours of search dwell time against GF5, there is no point $\boldsymbol{p}$ that can satisfy both constraint (4.4b) and constraint (4.4c) for interdictor 4 and GF5. The situation depicted illustrates that when the point $\boldsymbol{\theta}^{+}_{\mathrm{GF5}(1)}$ where search begins is contained in the outer circle, the point $\boldsymbol{\theta}^{-}_{\mathrm{GF5}(1)}$ where search ends cannot be contained in the inner circle.

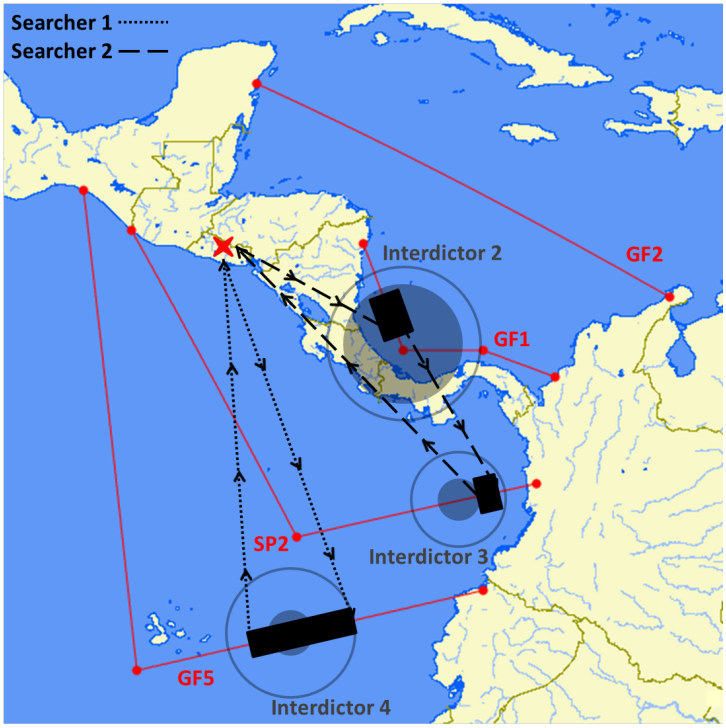


Figure 4.10: **Interdictor infeasibility of the restricted baseline scenario BS-FR solution** - The search boxes depicted represent the optimal search plan for **BS-FR**. The large transparent outlined outer circles represent the response range based on constraints (4.4b). The small shaded inner circles represent the response range based on constraints (4.4c). Given 4.64 hours of search dwell time against GF5, there is no point $\boldsymbol{p}$ that can satisfy both of these constraints for interdictor 4 and GF5. The situation depicted here shows that when the point $\boldsymbol{\theta}^{+}_{\mathrm{GF5}(1)}$ where search begins is contained in the outer ring, the point $\boldsymbol{\theta}^{-}_{\mathrm{GF5}(1)}$ where search ends cannot be contained in the inner ring.

Alternatively, one might consider using the optimal searcher path $\boldsymbol{x}^{*}$ for **BS-FR** and

solving a single NLP for all continuous variables in **SSP-I**. Such a strategy is also not guaranteed to yield a feasible solution. This is easily seen in a situation where at least two searched targets in the same interdiction zone are too far apart. In such a situation there would be no interdictor position $\boldsymbol{p}$ that could support searching both targets. In **BS-FR**, however, since no more that one target in each interdiction zone is searched in the optimal plan, we can obtain an interdictor-feasible plan using this strategy. This approach results in the search plan and interdictor placement given in Tables 4.12 and 4.13 respectively. This strategy produces a suboptimal plan, albeit by a small margin, with an expected search value that is 2.1 kg less than the optimal plan (given in Tables 4.10 and 4.11). While the optimality gap in this case is small, we note that its magnitude is driven by the expected payload of q_{GF5}.

Figure 4.11 shows a spatial representation of this suboptimal search plan. We observe that the search region for GF5 in this plan is shifted to the northeast and is slightly smaller than the search region for GF5 Figure 4.9.

Searcher	Segment	t	a	d	$\boldsymbol{\theta}^+$	$\boldsymbol{\theta}^-$
1	Home	-	0	10.75	(89.1W 13.4N)	(89.1W 13.4N)
	GF5(1)	2.66	13.40	4.20	(85.8W 0.6S)	(89.2W 1.4S)
	Home	2.73	20.33	-	(89.1W 13.4N)	(89.1W 13.4N)
2	Home	-	0	6.39	(89.1W 13.4N)	(89.1W 13.4N)
	GF1(3)	1.37	7.76	1.91	(82.2W 10.6N)	(82.8W 12.0N)
	SP2(1)	1.57	11.24	2.69	(78.5W 4.7N)	(79.2W 4.5N)
	Home	2.46	16.39	-	(89.1W 13.4N)	(89.1W 13.4N)

Table 4.12: **Optimal search times and searcher path using the optimal searcher path $\boldsymbol{x}$ for BS-FR** - Expected value of the search plan is 2,187.8 kg cocaine detected.

Interdictor	Position
1	-
2	(82W 10N)
3	(79.9W 4.1N)
4	(88.2W 1.1S)

Table 4.13: **Interdictor standby positions using the optimal searcher path $\boldsymbol{x}^*$ for BS-FR** - Interdictor 1 is not used in the optimal plan because GF2 is not searched.

4.6 Multiple Mission Cycles

We now consider search over the course of multiple mission cycles with a minimum recovery time period in between cycles. Similar multi-period sequencing models have been

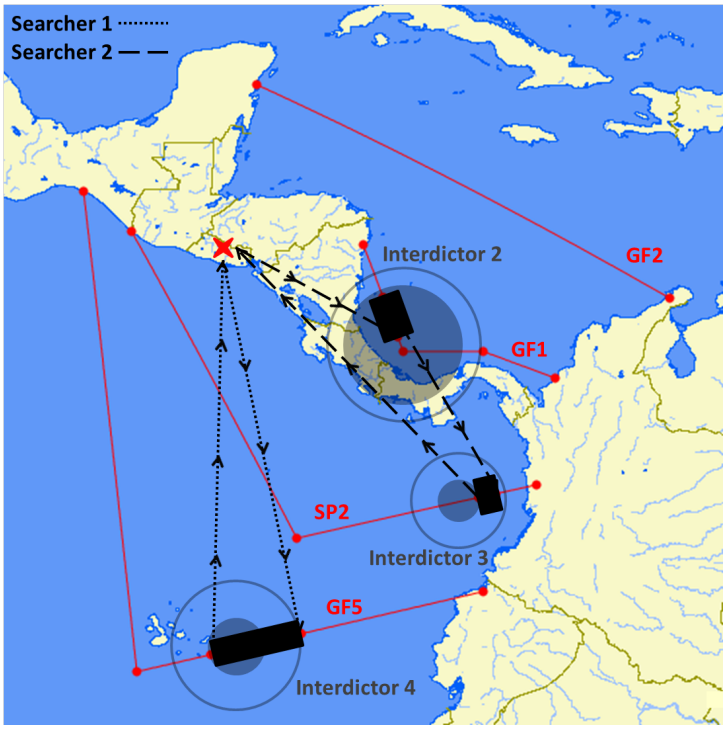


Figure 4.11: **Suboptimality of the restricted baseline scenario BS-FR fixed search order when adding interdictors** - The search boxes depicted represent the optimal search times for the **BS-FR** fixed search order. The large transparent outlined outer circles represent the response range based on constraints (4.4b). The small shaded inner circles represent the response range based on constraints (4.4c). Adding interdictor coordination to the fixed search order for **BS-FR** yields a plan whose objective function is 2.1 kg less than the optimal plan. This because the interdictor range restriction drives a slightly smaller search region for GF5.

studied in the vast literature on production planning and scheduling (see, e.g., Escudero and Salmeron 2005, Wolsey 1997, Dillenberger 1994). In another study, Yang et al. (2004) considers an receding horizon optimal control model that balances the short- and long-term rewards of routing a team of searchers. The focus here is on modeling optimal search problems of the form **SSP** over mutiple mission cycles. One can think of a mission cycle as being a day, though this particular unit of time need not be relevant in general. Planning for more than one mission cycle is particularly important when target movement tracks span multiple mission cycles (e.g., a daily mission cycle with target tracks that have $\tau_j^{max} - \tau_j^{min} > 24$ hours). In this situation, a multiple mission cycle scenario cannot simply be decoupled and solved one cycle at a time without considering how to address double-counting dwell times for these targets. One could choose to use a *myopic approach* where **SSP** is solved one cycle at a time by conditioning on the dwell time of undetected targets which were searched in a prior mission cycle. The focus of this sec-

tion is a more robust *multi-cycle approach*, where coordinated search plans for all mission cycles are computed prior to the start of the first cycle.

A multi-cycle approach has three potential advantages over the myopic approach. The first is that we obtain plans with greater search value by accounting for targets whose movement tracks span multiple mission cycles. This approach anticipates the benefit, if it exists, of choosing not to search for a target in an early cycle in favor of searching for it in a later cycle. Secondly, searcher recovery time between cycles is explicitly modeled. This allows planners to account for considerations like aircraft downtime due to maintenance checks and crew rest as part of the search plan, thus avoiding a situation where the myopic approach is unnecessarily constrained in later cycles by an unavailable searcher at the start of a cycle. Finally, when interdictor coordination is required, since interdiction assets may not be repositioned every day, the myopic approach would severely limit future cycle search plans based on where interdiction assets are placed in the first cycle.

There are two drawbacks to the multi-cycle approach. The first, as we will see shortly, is that we require a new model that is, strictly speaking, not just a special case of the GOP-RDR. Rather, it is a special case of a GOP-RDR that allows the vehicle to take multiple trips back the home station. The second drawback, of course, is that the multi-cycle model will be difficult to solve to optimality in a reasonable amount of time when many targets, multiple searchers, and several cycles are involved. Recall from Section 3.3 that Algorithm B&B applied to SSP performs efficiently because the perceived depth of the enumeration tree is shallow for large problems. The searcher runs out of endurance and fathoming occurs before B&B must extend the partial path very far. We now consider a multi-cycle SSP that resets the searchers remaining endurance time with each new cycle. We should expect a greater reliance on heuristic algorithms when considering multi-cycle search planning for scenarios like **BS**.

We define $C = \{1, 2, \ldots, c\}$ as the ordered set of mission planning cycles. We also define $H \subseteq N$ as the set of home station (or recovery) nodes, and $\hat{H} = H \backslash \{0, n+1\}$ as the set of *intermediate home station nodes* which exclude the originating and final destination home stations. For notational convenience we consider H to be an ordered set $\{h_0, h_1, \ldots, h_c\}$, where $h_0 = 0$ is the originating home station, $h_c = n + 1$ is the final destination home station, and there exists unique $(j, k) : h_k = j, j \in N \backslash \{0, n+1\}, k \in C \backslash \{0, c\}$ is the intermediate home station which completes cycle k. Let T_R be the minimum required

time between cycles for the searcher to recover. We denote by τ_j^{start} the nominal start time of cycle k, where $h_{k-1} = j$. We introduce the vector $\boldsymbol{z}$ of binary *cycle* variables with elements $z_{k,j}$ which take value 1 when region j is searched in cycle k, and 0 otherwise. We assume that each search region will be visited by the searcher at most once during the multi-cycle search plan. We recall that the set of target nodes, excluding home station nodes, is denoted by $\hat{N} = N \backslash H$. The multi-cycle SSP is stated as follows.

Problem SSP-C:

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t},\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \quad \sum_{j\in\hat{N}} q_j \left(1 - \exp\left\{-\alpha_j d_j y_j\right\}\right) \tag{4.5a}$$

$$\text{s.t.} \quad \sum_{j\in\hat{N}} d_j z_{k,j} + \sum_{(i,j)\in A} t_{i,j} z_{k,i} z_{k,j} \leq T, \qquad \forall k \in C \tag{4.5b}$$

$$d_j \geq T_R, \qquad \forall j \in \hat{H} \tag{4.5c}$$

$$a_j + d_j \geq \tau_j^{start}, \quad \forall j \in \hat{H} \tag{4.5d}$$

$$a_{h_1} \leq \ldots \leq a_{h_{c-1}} \leq a_{h_c} \tag{4.5e}$$

$$\sum_{k\in C} z_{k,j} - y_j = 0, \qquad \forall j \in \hat{N} \tag{4.5f}$$

$$z_{k,h_{k-1}} = z_{k,h_k} = 1, \qquad \forall k \in C \tag{4.5g}$$

$$z_{k,j} \in \{0,1\}, \quad \forall k \in C, j \in N \tag{4.5h}$$

(3.2b), (3.2c), and
(3.2e) – (3.2m)

The objective function (4.5a) computes the expected value of the search, excluding all home station nodes. Constraints (4.5b) enforce the searcher endurance resource constraint in each cycle. Constraints (4.5c) maintain the minimum recovery time T_R between cycles. Constraints (4.5d) enforce nominal start times of cycles, other than the first cycle which begins at time 0 by definition, by requiring that the searcher leave intermediate home stations no earlier than times τ_j^{start}. Constraints (4.5e) ensure that home stations are visited in the correct sequence. Constraints (4.5f) ensure that each target is searched at most once during the scenario. Constraints (4.5g) require that home stations are visited in their correct cycle. We observe that, since intermediate home stations lie on the boundary

between cycles, we model them as being in both the cycle immediately before and the cycle immediately after the searcher visits them. Constraints (4.5h) require that the cycle variables are binary.

We observe that constraints (4.5d) allow planners to maintain temporal integrity of planning cycles so that there is an opportunity for re-planning between cycles when the searcher is at the home station. In particular, the nominal start times τ_j^{start} may correspond to regularly spaced (e.g., 24-hour) time intervals so that the beginning of each mission execution cycle is aligned to the planning cycle. In our baseline scenario **BS**, the times τ_j^{start} correspond to 24-hour blocks of time. In principle this constraint could be relaxed by setting $\tau_j^{start} = 0, \forall j \in \hat{H}$. While this would yield a more flexible model where cycles can slide throughout the mission execution period $[0, D]$, re-planning between cycles may not be possible.

We assume paths $\boldsymbol{x}$ are generated so that constraints (4.5e) through (4.5h) are satisfied. Formally, we define the set $\mathbb{X}^C \subseteq \mathbb{X}$ as the subset of paths that satisfy cycle constraints (4.5e) through (4.5h). We observe that this restriction on paths $\boldsymbol{x}$ only means that home station nodes must have the proper ordering in the path (4.5e) and that they must all be visited (4.5g). This reduces the number of partial paths that must be considered in the modified branch-and-bound algorithm described in Section 4.6.1, and path enumeration can be accomplished with a simple adjustment to the branching rules. We denote by $N_k(\boldsymbol{x})$ and $A_k(\boldsymbol{x})$ the set of nodes and the set of arcs respectively that are in cycle k in the path $\boldsymbol{x}$. Analogously to **SSP($\boldsymbol{x}$)**, we define the fixed path multi-cycle search model for any $\boldsymbol{x} \in \mathbb{X}^C$ as follows.

Problem SSP-C($\boldsymbol{x}$):

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t},\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \quad \sum_{j \in \hat{N}(\boldsymbol{x})} q_j \left(1 - \exp\left\{-\alpha_j d_j y_j\right\}\right) \tag{4.6a}$$

$$\text{s.t.} \quad \sum_{j \in \hat{N}_k(\boldsymbol{x})} d_j + \sum_{(i,j) \in A_k(\boldsymbol{x})} t_{i,j} \leq T, \qquad \forall k \in C \tag{4.6b}$$

(3.2h), (3.2i), (3.3b), (3.3c),
(3.3e) – (3.3i), (4.5c), and (4.5d)

4.6.1 Algorithms for the Multi-Cycle Model

The optimal solution to **SSP-C** can be computed using Algorithm B&B, where the relaxed problem solved in step 2 is modified to account for multiple mission cycles as follows.

We denote by $\hat{N}_k(\hat{\boldsymbol{x}}_\ell)$ the set of target nodes in the partial path $\hat{\boldsymbol{x}}_{\boldsymbol{\ell}}$ that are visited in cycle k. Recall that the set of target nodes in the partial path is denoted by $\hat{N}(\hat{\boldsymbol{x}}_\ell)$. The set of arcs in the partial path $\hat{\boldsymbol{x}}_{\boldsymbol{\ell}}$ that are transited in cycle k is defined as $A_k(\hat{\boldsymbol{x}}_{\boldsymbol{\ell}})$. We define the indicator parameter $\bar{I}_k$ which takes value 1 when cycle k is complete and 0 otherwise.

The smallest minimum travel time between any unvisited target node and the final destination home station is defined as follows.

$$\delta^*(\hat{\boldsymbol{x}}_{\boldsymbol{\ell}}) = \min_{j \in N \setminus \hat{N}(\hat{\boldsymbol{x}}_\ell)} \{\delta_{j,n+1}\}$$

Given that cycles completed in the partial path $\hat{\boldsymbol{x}}_{\boldsymbol{\ell}}$ are indicated by $\bar{I}_k$, the optimistic endurance time $T(\hat{\boldsymbol{x}}_{\boldsymbol{\ell}})$ that remains for all future cycles is computed as follows.

$$T(\hat{\boldsymbol{x}}_{\boldsymbol{\ell}}) = \left(c - 1 - \sum_{k'=2}^{c} \bar{I}_{k'}\right) \max\left\{0, T - 2\delta^*(\hat{\boldsymbol{x}}_{\boldsymbol{\ell}})\right\} \tag{4.7}$$

The term (4.7) is an upper bound on the total search dwell time of future mission cycles which are explored by extending the current partial path $\hat{\boldsymbol{x}}_{\boldsymbol{\ell}}$. Just as in constraint (3.6c), searcher endurance T is decremented by twice the minimum travel time between home and any potential target. In this case, however, we account for the fact that there may be more cycles beyond the current one that are yet to be explored by extending the partial path.

In order to have a useful bound on transit time when the last node ℓ on the partial path is a home station node, we define the home-node-dependent minimum travel time as follows.

$$\tilde{\delta}_\ell = \begin{cases} 2\delta^*(\hat{\boldsymbol{x}}_{\boldsymbol{\ell}}), & \ell \in H \\ \delta_{\ell,n+1}, & \ell \notin H \end{cases}$$

We define the indicator parameter $\hat{I}_k$, which takes value 1 when cycle k is in progress and

incomplete (e.g., when the partial path $\hat{\boldsymbol{x}}_\ell$ has reached home station node h_{k-1} but not home station node h_k), and 0 otherwise. The partial path relaxation model is stated as follows.

Problem RSSP-C($\hat{\boldsymbol{x}}_\ell$):

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t}} \quad \sum_{j\in\hat{N}} q_j\left(1-\exp\left\{-\alpha_j d_j\right\}\right) \tag{4.8a}$$

$$\text{s.t.} \quad \underbrace{\sum_{j\in\hat{N}_k(\hat{\boldsymbol{x}}_\ell)} d_j}_{\text{(i)}} + \underbrace{\hat{I}_k\left(\sum_{j\in\hat{N}\setminus\hat{N}(\hat{\boldsymbol{x}}_\ell)} d_j\right)}_{\text{(ii)}}$$

$$\ldots + \underbrace{\sum_{(i,j)\in A_k(\hat{\boldsymbol{x}}_{\boldsymbol{\ell}})} t_{i,j}}_{\text{(iii)}} \leq \underbrace{T-\hat{I}_k\tilde{\delta}_\ell}_{\text{(iv)}} + \underbrace{\hat{I}_k T(\hat{\boldsymbol{x}}_{\boldsymbol{\ell}})}_{\text{(v)}}, \quad \forall k\in C \tag{4.8b}$$

$$d_j \geq T_R, \qquad \forall j\in\hat{H}\cap N(\hat{\boldsymbol{x}}_\ell) \tag{4.8c}$$

$$a_j + d_j \geq \tau_j^{start}, \qquad \forall j\in\hat{H}\cap N(\hat{\boldsymbol{x}}_\ell) \tag{4.8d}$$

$$\text{(3.2f) – (3.2j), (3.6b), (3.6f),}$$
$$\text{(3.7b), (3.7c), and (3.7e) – (3.7i)}$$

We note that when $\ell = n+1$ (or equivalently $I_\ell = 1$), the path is complete, and **SSP-C($\boldsymbol{x}$)** and **RSSP-C($\hat{\boldsymbol{x}}_\ell$)** are equivalent. Constraints (4.8b) are a relaxation of (4.5b) and become successively restricted as the current partial path $\hat{\boldsymbol{x}}_\ell$ is extended. The meaning of each term in this constraint is as follows. Term (i) sums dwell time of nodes visited in cycle k. Term (ii) sums dwell time for target nodes that are not in the partial path when cycle k is in progress and incomplete. Term (iii) sums transit time in cycle k. Term (iv) decrements the searcher endurance by the minimum travel time. Term (v) adds the optimistic total mission time for future cycles when cycle k is in progress and incomplete. We observe that when cycle k is complete, constraint (4.8b) takes the form of (4.6b), and when cycle k is not yet in progress constraint (4.8b) is vacuous. We assume branching is performed so that constraints (4.5e)-(4.5h) are satisfied throughout.

Though similar, **RSSP-C($\hat{\boldsymbol{x}}_\ell$)** does not possess the same structure as **RSSP($\hat{\boldsymbol{x}}_\ell$)** due

to the handling required for multiple mission cycles. Therefore, the following results are warranted to justify using Algorithm B&B for **SSP-C**.

Lemma 1. *The set* $\Phi = \{(a, b, c, c', d, e) \in \mathbb{R}^6 : a + c \leq d;\ b \leq e;\ c \geq c'\}$ *is a restriction of the set* $\Phi' = \{(a, b, c, c', d, e) \in \mathbb{R}^6 : a + b + c' \leq d + e;\ c \geq c'\}$.

Proof. Given $(a, b, c, c', d, e, e') \in \Phi$, the result follows by observing that $c \geq c'$ and adding the first two constraints in Φ together.

$$a + b + c' \ \leq \ a + c + b \ \leq \ d + e$$

□

Theorem 2. **RSSP-C($\hat{\boldsymbol{x}}_{\ell'}$)** *is a restriction of* **RSSP-C($\hat{\boldsymbol{x}}_{\ell}$)**, $\forall \ell' : (\ell, \ell') \in A$.

Proof. Using Theorem 1 and observing that constraints (4.8c) are unchanged by partial path extensions, it suffices to show that whenever variables $\boldsymbol{a}, \boldsymbol{d}$, and $\boldsymbol{t}$ are feasible in **RSSP-C($\hat{\boldsymbol{x}}_{\ell'}$)**, they also satisfy (4.8b) in **RSSP-C($\hat{\boldsymbol{x}}_{\ell}$)**. There are three cases to consider.

Case 1: $\ell' \in \hat{N}$. Adding node ℓ' to the partial path does not complete a cycle. Suppose the current, yet-to-be completed cycle is k'. Constraints (4.8b) where $k \neq k'$ are unchanged. In the current cycle constraint (4.8c) associated with k', $d_{\ell'}$ remains on the left hand side by shifting from term (ii) to term (i), $t_{\ell,\ell'}$ is added to term (iii), $\tilde{\delta}_\ell$ is replaced with $\delta_{\ell',n+1}$ in term (iv), and $T(\hat{\boldsymbol{x}}_{\boldsymbol{\ell}})$ is replaced with $T(\hat{\boldsymbol{x}}_{\boldsymbol{\ell'}})$ in term (v). Since $T(\hat{\boldsymbol{x}}_{\boldsymbol{\ell}}) - \tilde{\delta}_\ell \geq T(\hat{\boldsymbol{x}}_{\boldsymbol{\ell'}}) - t_{\ell,\ell'} - \delta_{\ell',n+1}$, the result follows.

Case 2: $\ell' \in \hat{H}$. Adding node ℓ' to the partial path completes a cycle and starts a new cycle. Suppose the cycle just completed is k' and that the new cycle that just began is $k' + 1$. Constraints (4.8b) where $k \notin \{k', k' + 1\}$ are unchanged. In the just-completed cycle constraint (4.8b) associated with k', $d_{\ell'}$ remains on the left hand side by shifting from term (ii) to term (i), $t_{\ell,\ell'}$ is added to term (iii), and the right hand side is reduced to T. Meanwhile all dwell times d_j for unvisited target nodes in term (ii) are moved to the new cycle constraint (4.8b) associated with $k' + 1$. Since $T(\hat{\boldsymbol{x}}_{\boldsymbol{\ell}}) = T - 2\delta^*(\hat{\boldsymbol{x}}_{\boldsymbol{\ell'}}) + T(\hat{\boldsymbol{x}}_{\boldsymbol{\ell'}})$ the result follows by Lemma 1.

Case 3: $\ell' = n + 1$. Adding node ℓ' to the partial path completes the path. Constraints (4.8b) where $k < c$ are unchanged. In the final cycle constraint (4.8b) associated with

$k = c$, $d_{\ell'}$ remains on the left hand side by shifting from term (ii) to term (i), $t_{\ell,\ell'}$ is added to term (iii), and the right hand side is reduced to T. Since all dwell times d_j for unvisited target nodes in term (ii) are constrained to 0, the result follows. □

Let $Z^C(\hat{\boldsymbol{x}}_\ell)^*$ be the optimal objective function value of **RSSP-C**$(\hat{\boldsymbol{x}}_\ell)$.

Corollary 2. $Z^C(\hat{\boldsymbol{x}}_\ell)^* \geq Z^C(\hat{\boldsymbol{x}}_{\ell'})^*, \forall \ell' : (\ell, \ell') \in A$.

Proof. If $\ell = n + 1$, then $\{\ell' : (\ell, \ell') \in A\} = \emptyset$. Alternatively, if $\ell \in N \backslash \{n + 1\}$ the result follows because **RSSP-C**$(\hat{\boldsymbol{x}}_{\ell'})$ is a restriction of **RSSP-C**$(\hat{\boldsymbol{x}}_\ell)$ by Theorem 2. □

4.6.2 Bounding Strategies

Since **SSP-C** can be difficult to solve in a reasonable amount of time for large problems with multiple cycles, it is important to highlight some strategies to compute upper bounds on its optimal objective function value. An upper bound is required to compute the relative optimality gap ($rgap$) of a heuristic solution. Indeed obtaining useful bounds and developing good heuristic algorithms are not mutually exclusive endeavors. Given an upper bound UB and objective function value of any feasible search plan Z, the relative optimality gap is computed as follows.

$$rgap = \frac{|UB - Z|}{UB} \times 100\% \tag{4.9}$$

The simplest upper bound is obtained directly from evaluating the value of the a do-nothing solution where the searcher merely remains at the home station. In this case the bound is $UB = \sum_{j \in \hat{N}} q_j$. While this is an uninteresting bound in and of itself, it is easy to compute and may lead to an adequate $rgap$ for some problem instances. We call this the *do-nothing bound*.

Another simple bound is obtained by solving the root node relaxation problem **RSSP(0)**. Clearly, this is better than the do-nothing bound because dwell time is limited by constraints (3.6b)-(3.6f). We call this the *root bound*. Computing this bound requires solving a single convex NLP, therefore, it is only slightly harder to obtain the root bound than it is to obtain the do-nothing bound.

Another upper bound is obtained by using the l-level PBB heuristic as discussed in Section 3.2.2. When $l = 0$, this approach is equivalent to computing the root bound. When l is large this approach may be too expensive to compute for large problem instances. Depending on the problem setting, useful bounds may be obtained in a reasonable amount of time. We call this the *l-level PBB bound*.

Since the complexity **SSP-MS** is exponential in the number of searchers $|S|$, assuming homogeneous searchers operating from the same home station, one could simply solve the single searcher problem and multiply the resulting search plan value by $|S|$. This is clearly an upper bound for the multi-searcher problem because we allow "double-counting" dwell time. We call this the *$|S|$-multiple bound*.

The final approach to obtain an upper bound on the optimal objective function value is to decouple the problem for each mission cycle. In this approach, for each cycle, we model any target whose movement track occurs at any time within the cycle, allow "double counting " dwell time, and make the searcher available at the start of the cycle. This approach clearly leads to an upper bound because "double-counting " dwell time is allowed and constraints (4.5c) are relaxed. We call this the *decoupled cycle bound*.

4.6.3 Multi-Cycle Model Numerical Results

We return to the full baseline scenario **BS** as we are now able to model search planning for both searchers and all nine targets over the full 72-hour period. We assume that the daily planning cycle entails that each searcher return to the home station by the end of the mission day and that they cannot leave the home station until the new mission day has begun. This gives planners the ability to adjust the plan, if desired, by re-solving between mission cycles. It should be noted that the terms "day k" and "cycle k" are interchangeable in this scenario. Given the 10-hour searcher endurance, the earliest arrival time τ^{min} to the home station is well defined as 10 hours after the beginning of the mission day. Similarly, latest departure time τ^{max} from the home station is well defined as 10 hours prior to the end of the mission day. Table 4.14 lists the resulting time windows associated with home station nodes.

Before discussing solutions to **BS**, we report upper bounds on the optimal objective function value using the approaches outlined in Section 4.6.2. Table 4.15 lists these bounds, along with computation time required to obtain each of them. We see that the

node	τ^{min}	τ^{start}	τ^{max}
h_0	0	-	14
h_1	10	24	38
h_2	34	48	62
h_3	58	-	72

Table 4.14: **Baseline scenario BS home station node time windows (hours)**

2-level PBB approach yields the smallest upper bound of 9,588 kg in a relatively short runtime of 438 seconds. The easy-to-compute root bound is only slightly higher at 9,925 kg. The more sophisticated bounding strategies, 2-multiple and decoupled cycle, are expensive to compute and yield bounds that are nearly as high as the do-nothing bound. For the remainder of this section we will use the 2-level bound in computing relative optimality gaps.

Bound Approach	UB (kg)	Runtime (sec)
do-nothing	14 500	<1
root	9925	<1
2-level PBB	9588	438
2-multiple	10 468	625
decoupled cycle	10 452	36 251

Table 4.15: **BS optimal search plan bounds** - Computed using bounding strategies described in Section 4.6.2. The lowest upper bound is obtained using 2-level PBB strategy. The more sophisticated bounding strategies, 2-multiple and decoupled cycle, are expensive to compute and yield poor bounds.

We now present numerical results for the full baseline scenario **BS** computed using the myopic appraoch (described at the beginning of Section 4.6), the SSP heuristic, the CS heuristic, and Algorithm B&B.

Search and Interdiction Plan Computed using the Myopic Approach

A search and interdiction plan could be computed using the myopic approach. Table 4.16 lists each active target segment by day based on its time window $[\tau^{min}, \tau^{max}]$. The myopic approach proceeds each day considering only active targets.

We know from Section 4.5.1 that the optimal first day plan, reported in Tables 4.10-4.11, directs the position of interdictors 2-4 and has an expected detection value of 2,189.8 kg. During the second day only the targets listed in Table 4.17 are active based on the time windows associated with their movement tracks. We condition on day 1 dwell times, denoted by $\bar{d}$ in the objective function (4.1a) as shown in (4.10).

Target	Day 1	Day 2	Day 3	τ^{min}	τ^{max}
GF1(1)	X			2.00	3.46
GF1(2)	X			3.46	7.06
GF1(3)	X			7.06	10.18
SP1		X	X	32.00	59.06
GF2	X	X	X	20.93	68.93
MV1(1)		X	X	41.00	63.85
MV1(2)			X	63.86	72.00
GF3			X	52.00	70.37
GF4(1)		X	X	47.00	51.80
GF4(2)			X	51.80	56.75
GF4(3)			X	56.75	71.78
SP2(1)	X	X		8.00	41.88
SP2(2)		X	X	41.88	72.00
SP3		X	X	28.90	72.00
GF5(1)	X			7.00	21.01
GF5(2)	X	X		21.01	40.74

Table 4.16: **Active target segments for each day** - Active targets for each day are marked "X". Time windows $[\tau^{min}, \tau^{max}]$ are derived from the scenario **BS** description in Section 4.1.

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t},\boldsymbol{x},\boldsymbol{y}} \sum_{i \in F} q_i \left(1 - \exp\left\{ - \sum_{j \in B(i)} \alpha_j (d_j + \bar{d}_j) y_j \right\} \right) \tag{4.10}$$

Target	Day 1 Dwell Time $\bar{d}$
SP1	0
GF2	0
MV1(1)	0
GF4(1)	0
SP2(1)	2.69
SP2(2)	0
SP3	0
GF5(2)	4.24

Table 4.17: **Active target segments for myopic approach day 2** - Dwell times shown reflect optimal day 1 solution from Table 4.10.

Since interdictors are not repositionable from one day to the next, we fix the location of interdictors 2-4 to the positions given in Table 4.11. This is clearly a significant restriction on search plans considered after the first day. This requirement could be relaxed by allowing interdictor position $\boldsymbol{p}$ to change with each new cycle or by allowing drift vector

$\boldsymbol{v}$ to take nonzero values, however we enforce this restriction due to the high cost of repositioning surface assets. Additionally, we enforce recovery time constraint (4.5c) by placing a lower bound on d_0 as required. The resulting myopic search plan for day 2 is given in Table 4.18 and Figure 4.12.

Searcher	Segment	t	a	d	$\boldsymbol{\theta}^+$	$\boldsymbol{\theta}^-$
1	Home	-	16.39	27.26	(89.1W 13.4N)	(89.1W 13.4N)
	SP1	1.73	45.39	0.92	(79.9W 11.5N)	(80.1W 11.6N)
	Home	1.70	48.00	-	(89.1W 13.4N)	(89.1W 13.4N)
2	Home	-	24.00	19.65	(89.1W 13.4N)	(89.1W 13.4N)
	SP1	1.73	45.39	0.92	(79.9W 11.5N)	(80.1W 11.6N)
	Home	1.70	48.00	-	(89.1W 13.4N)	(89.1W 13.4N)

Table 4.18: **Myopic approach day 2 search times and searcher path** - High-value target SP2 is assigned to both searchers but dwell time is limited by position of interdictor 2 which is fixed by the day 1 solution.

Figure 4.12: **Myopic approach day 2 solution** - Both searchers are assigned to search SP1. Search against high-value target SP1 is limited by the position of interdictor 2, which is fixed in day 1 to support searching GF1.

In Figure 4.12 we see that the position of interdictor 2, which was fixed in day 1 to support searching GF1, limits search opportunity against the highest value target SP1 in day 2.

This is clearly an undesirable search plan. While it is good that the highest value target, SP1, is searched, a more robust plan would have positioned interdictor 2 closer to SP1. Interdictor 1 is unused through day 2, and GF2 and MV1 are not searched in day 2.

Similarly, we condition on active day 3 targets (see Table 4.19) using (4.10), restrict the position of interdictors 2-4 to day 1 locations, and enforce recovery constraint (4.5c) to obtain the conditioned optimal search and interdiction plan for day 3 shown in Tables 4.20 and 4.21, and Figure 4.13. The only remaining untasked interdictor, interdictor 1, is positioned to support search against both MV1 and GF3. A good opportunity to search high-value target, SP1, when it is close to the home station is wasted because interdictor 2 is not in range. Therefore, SP1 is not searched in day 3.

Target	Days 1 & 2 Dwell Time $\bar{d}$
SP1	1.84
GF2	0
MV1(1)	0
MV1(2)	0
GF3	0
GF4(1)	0
GF4(2)	0
GF4(3)	0
SP2(2)	2.69
SP3	0

Table 4.19: **Active targets for myopic approach day 3** - Dwell times shown reflect the sum of optimal dwell times from Tables 4.10 and 4.17.

Searcher	Segment	t	a	d	$\boldsymbol{\theta}^+$	$\boldsymbol{\theta}^-$
1	Home	-	48.00	6.00	(89.1W 13.4N)	(89.1W 13.4N)
	MV1(1)	0.54	58.71	1.76	(77.5W 15.8N)	(77.7W 16.2N)
	GF3	2.02	56.02	2.15	(78.1W 12.6N)	(79.5W 13.7N)
	Home	2.16	62.63	-	(89.1W 13.4N)	(89.1W 13.4N)
2	Home	-	48.00	6.00	(89.1W 13.4N)	(89.1W 13.4N)
	MV1(1)	0.54	58.71	1.76	(77.5W 15.8N)	(77.7W 16.2N)
	GF3	2.02	56.02	2.15	(78.1W 12.6N)	(79.5W 13.7N)
	Home	2.16	62.63	-	(89.1W 13.4N)	(89.1W 13.4N)

Table 4.20: **Myopic approach day 3 search times and searcher path** - Both searchers are assigned to MV1 and GF3 which correspond to the only remaining untasked interdictor. Each is limited by the 6-hour recovery time at home station and cannot start their search plan until time 54 hours.

The myopic approach yields total search dwell times for each target listed in Table 4.22,

Interdictor	Position
1	(78.4W 15.2N)
2	(81.9W 10N)*
3	(79.9W 4.1N)*
4	(91.0W 1.7S)*

Table 4.21: **Myopic approach day 3 optimal interdictor positions** - Positions of interdictors 2-4 (marked *) are fixed to the day 1 positions. Interdictor 1 position is set to support search of MV1 and GF3 on day 3.

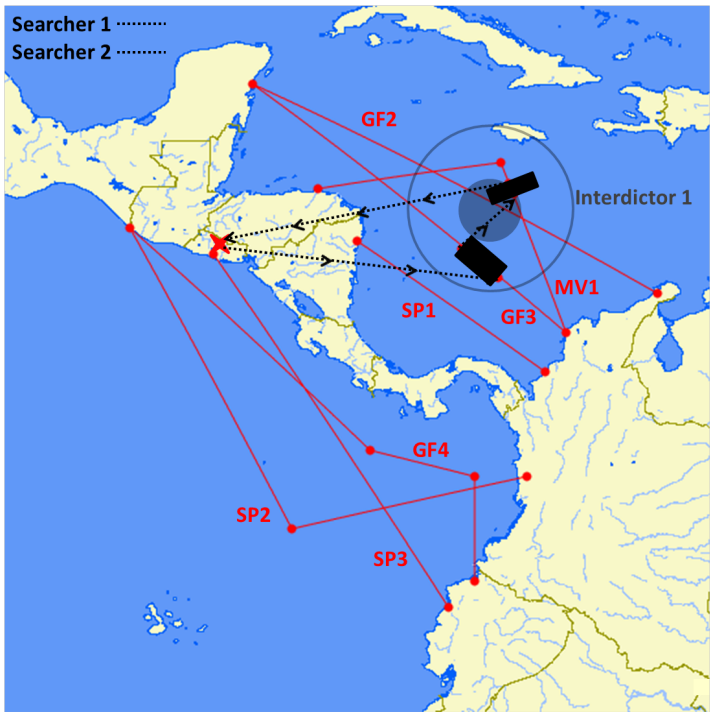


Figure 4.13: **Myopic approach solution day 3** - Interdictor 1, untasked through day 2, is positioned to support search against MV1 and GF3.

leading to a total expected search value of 4,469.2 kg. Recall that the upper bound on the true optimal objective function value is 9,588 kg. Therefore the myopic approach solution, computed in a 480.4 seconds (8.0 minutes) of total runtime, has an *rgap* of 53%.

Search and Interdiction Plan Computed using the SSP Heuristic

Using the time windows given in Table 4.14, the SSP heuristic, with two modifications, can be used to find a good, though not necessarily optimal, search plan in **SSP-C**. The first modification is that intermediate home stations cannot be removed from the heuristic path $\bar{\boldsymbol{x}}$ in any phase of the heuristic. The second is that, for the purpose of assigning orderings O_j to intermediate home station nodes in the heuristic, we use the τ_j^{start} instead

Target	Total Dwell Time
GF1	1.91
SP1	1.84
GF2	0
MV1	3.52
GF3	4.30
GF4	0
SP2	2.69
SP3	0
GF5	4.24

Table 4.22: **Myopic approach total search dwell times**

of τ_j^{max}. This provides a fairly equal spacing in the heuristic search order, placing at least two target segments in each mission cycle.

Tables 4.23 and 4.24 list the SSP heuristic search plan and interdictor positioning respectively. This plan yields and expected search value of 6,380.5 kg ($rgap = 33\%$) and is computed in 3.1 seconds. While this solution is clearly much better and computed more quickly than the myopic solution, it can certainly be improved. We observe that both searchers remain at the home station throughout the second day - a *do-nothing cycle*. Additionally, interdictor 3 is unused throughout the scenario. Clearly there would be some benefit to searching, for example, SP2 and/or SP3 during day 2. These targets are removed from the heuristic search order early in step 8 because their rate-of-reward $\hat{\beta}$ is relatively low. While this exposes a drawback of this linear-time heuristic applied to multi-cycle problems, it is important to note that the high-value targets, SP1 and GF5, are heavily searched.

Search and Interdiction Plan Computed using the CS Heuristic

While the SSP heuristic solution is much better than the myopic approach solution, it is still far from optimal and a manual examination of this plan highlights some opportunities for improvement. We now consider using the CS heuristic described in Section 4.3.1. Using cycle limit parameter $\Lambda = 3$, we obtain the search plan and interdictor positioning given in Tables 4.25 and 4.26. The CS heuristic solution has a search plan value of 7,450.1 kg ($rgap = 22\%$) and was obtained in 1,786.3 seconds (29.8 minutes).

The CS heuristic solution has several remarkable aspects. First, the searcher paths $\boldsymbol{x}_s$ differ from the SSP heuristic plan only in day 2. The CS heuristic solution adds search

Searcher	Segment	t	a	d	$\boldsymbol{\theta}^+$	$\boldsymbol{\theta}^-$
1	Home h_0	-	0	7.99	(89.1W 13.4N)	(89.1W 13.4N)
	GF5(1)	2.69	10.68	4.18	(83.6W 0.1S)	(87.0W 0.9S)
	Home h_1	2.66	17.52	16.48	(89.1W 13.4N)	(89.1W 13.4N)
	Home h_2	0	34.00	16.24	(89.1W 13.4N)	(89.1W 13.4N)
	SP1	1.47	51.71	7.35	(81.2W 12.4N)	(82.7W 13.4N)
	Home h_3	1.18	60.24	-	(89.1W 13.4N)	(89.1W 13.4N)
2	Home h_0	-	0	5.93	(89.1W 13.4N)	(89.1W 13.4N)
	GF1(3)	1.42	7.35	2.38	(82.1W 10.2N)	(82.8W 12.1N)
	GF5(1)	2.32	12.05	1.23	(84.7W 0.3S)	(85.7W 0.6S)
	Home h_1	2.66	15.93	18.07	(89.1W 13.4N)	(89.1W 13.4N)
	Home h_2	0	34.00	18.82	(89.1W 13.4N)	(89.1W 13.4N)
	SP1	1.37	54.18	4.88	(81.7W 12.7N)	(82.7W 13.4N)
	GF3	0.45	59.5	1.88	(80.4W 14.4N)	(81.6W 15.4N)
	Home h_3	1.41	62.82	-	(89.1W 13.4N)	(89.1W 13.4N)

Table 4.23: **Multi-cycle SSP heuristic searcher solution** - High-value targets SP1 and GF5 are heavily search by both searchers. The day 2 portion of the plan, a do-nothing cycle, could clearly be improved.

Interdictor	Position
1	(81.3W 15.1N)
2	(82.6W 13.0N)
3	-
4	(86.0W 0.6S)

Table 4.24: **Multi-cycle SSP heuristic interdictor solution** - Interdictor 1 is positioned to support search against GF3 in day 3. Interdictor 2 is positioned to support search against GF1 in day 1 and SP1 in day 3. Interdictor 3 is unused. Interdictor 4 is positioned to support search against GF5 in day 1.

against SP2 in day 2 for both searchers. The absence of a day 2 search assignment in the SSP heuristic solution was its most glaring weakness. The CS heuristic solution corrects this problem. Second, all interdiction assets are used. Since interdictor range restricts search opportunities, it is reasonable to conclude that there is little room for improvement in this plan. Third, high-value targets SP1 and GF5 are both searched heavily. Fourth, each day is characterized by targets being searched by both searchers. Since interdictor range limits search opportunity and there are only four interdiction assets, it is reasonable that both searchers would focus on the most valuable targets each day so that there are interdictors left to support searching high-value targets on other days.

Figure 4.14 depicts day 1 of the CS heuristic solution. We see that GF1 is searched by

Searcher	Segment	t	a	d	$\boldsymbol{\theta}^+$	$\boldsymbol{\theta}^-$
	Home h_0	-	0	7.99	(89.1W 13.4N)	(89.1W 13.4N)
	GF5(1)	2.69	10.68	4.18	(83.6W 0.1S)	(87.0W 0.9S)
	Home h_1	2.66	17.52	16.36	(89.1W 13.4N)	(89.1W 13.4N)
1	SP2(1)	2.04	35.93	5.95	(84.6W 3.3N)	(86.0W 3.0N)
	Home h_2	2.01	43.89	6.35	(89.1W 13.4N)	(89.1W 13.4N)
	SP1	1.47	51.71	7.35	(81.2W 12.4N)	(82.7W 13.4N)
	Home h_3	1.18	60.24	-	(89.1W 13.4N)	(89.1W 13.4N)
	Home h_0	-	0	5.93	(89.1W 13.4N)	(89.1W 13.4N)
	GF1(3)	1.42	7.35	2.38	(82.1W 10.2N)	(82.8W 12.1N)
	GF5(1)	2.32	12.05	1.23	(84.7W 0.3S)	(85.7W 0.6S)
	Home h_1	2.66	15.93	17.95	(89.1W 13.4N)	(89.1W 13.4N)
2	SP2(1)	2.04	35.93	5.95	(84.6W 3.3N)	(86.0W 3.0N)
	Home h_2	2.01	43.89	8.93	(89.1W 13.4N)	(89.1W 13.4N)
	SP1	1.37	54.18	4.88	(81.7W 12.7N)	(82.7W 13.4N)
	GF3	0.45	59.5	1.88	(80.4W 14.4N)	(81.6W 15.4N)
	Home h_3	1.41	62.82	-	(89.1W 13.4N)	(89.1W 13.4N)

Table 4.25: **CS heuristic searcher solution to BS** - The searcher paths $\boldsymbol{x}_s$ differ from the SSP heuristic plan only in day 2. The CS heuristic solution adds search against SP2 in day 2 for both searchers.

Interdictor	Position
1	(81.3W 15.1N)
2	(82.6W 13.0N)
3	(86.0W 3.0N)
4	(86.0W 0.6S)

Table 4.26: **CS heuristic interdictor solution to BS** - Interdictor 1 is positioned to support search against GF3 in day 3. Interdictor 2 is positioned to support search against GF1 in day 1 and SP1 in day 3. Interdictor 3 is positioned to support search against SP2 in day 2. Interdictor 4 is positioned to support search against GF5 in day 1.

searcher 2 in a area that places interdictor 2 in position to be able to support later search of SP1 in day 3. Searcher 1 spends its entire day 1 mission assigned to GF5. We observe that the dwell time of seacher 1 against GF5 is limited by the range of interdictor 4.

Figure 4.15 depicts day 2 of the CS heuristic solution. Both searchers are assigned to SP2 and interdictor 3 is positioned to support. In the day 2 solution, dwell time against SP2 is not limited by interdictor range. Both searchers reach their endurance limit T during day 2.

The CS heuristic solution for final day of the scenario is depicted in Figure 4.16. High-

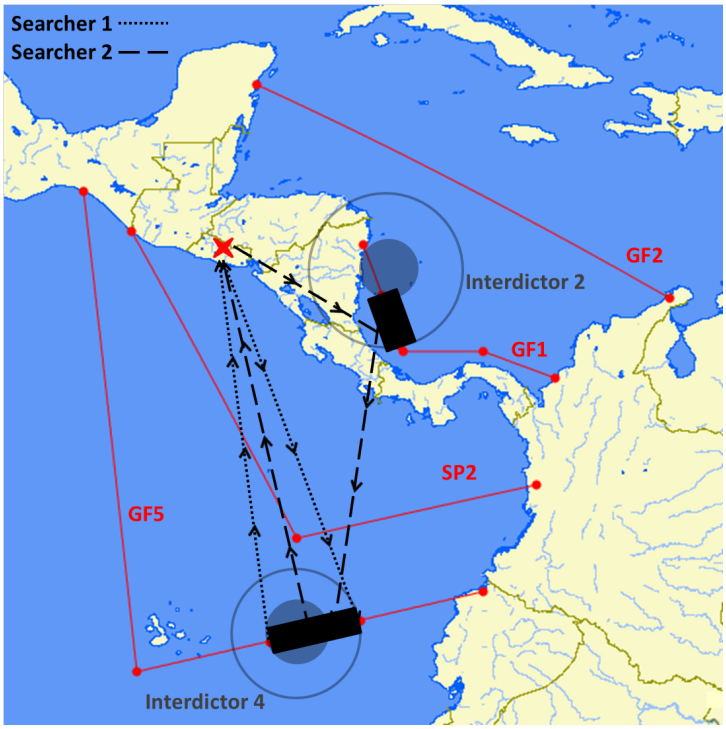


Figure 4.14: **CS heuristic solution to BS day 1** - GF1 is searched by searcher 2 in a area that places interdictor 2 (upper circles) in position to be able to support later search of SP1 in day 3. Searcher 1 spends its entire day 1 mission assigned to GF5. Dwell time of seacher 1 against GF5 is limited by the range of interdictor 4 (lower circles). Response circles for interdictor 4 relative to searcher 2 not shown.

value target SP1 is searched as its track is closest to the home station, allowing transit time to be small. Searcher 2 searches SP1 and then searches GF3. Having spent much of its endurance limit searching against SP1, little time is left to search against GF3. This results in a small response circle for interdictor 1.

Search and Interdiction Plan Computed using Algorithm B&B

The optimal solution to **BS**, computed using Algorithm B&B with RPPE in 24,947.9 seconds (6.9 hours), has search plan value of 7,851.4 kg. While the two-hour time limit is exceeded, with greater computing power it may be possible to obtain the optimal solution in a period of time that can support a 24-hour planning cycle. Tables 4.27 and 4.28 list the optimal search plan and interdiction plan respectively.

The optimal solution to **BS** has several features that distinguish it from the heuristic solutions. First, in the optimal solution, both searchers are assigned to the same targets through the first two days of the scenario. In this case, the strategy of focusing both

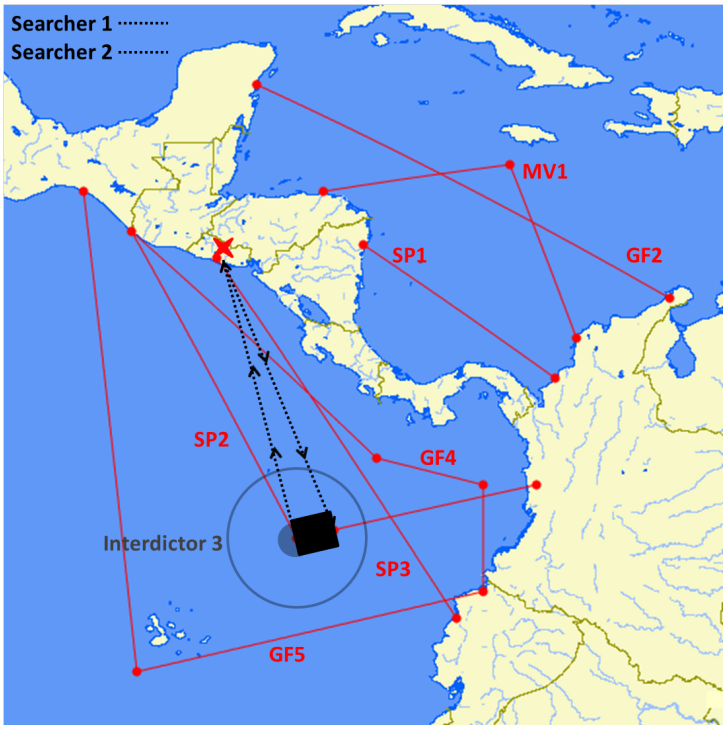


Figure 4.15: **CS heuristic solution to BS day 2** - Both searchers are assigned to SP2 and interdictor 3 is positioned to support. Dwell time against SP2 is limited by searcher endurance, not interdictor range.

searchers on high-value targets is preferable given the limited range of interdiction assets. Second, high-value target SP1 is searched at the very end of day 2, rather than on the day 3 as the CS heuristic solution prescribes. This allows GF4 to be added to the search plan in the third day. GF4 is not searched in any of the heuristic solutions. Third, the optimal plan in days 2 and 3 of the scenario is marked by searcher paths that occur as late as possible. This suggests that the optimal plan is limited by the time windows on the intermediate home stations (given in Table 4.14). Shifting the time windows by adding four hours to the latest home station departure times $\tau_j^{max}, \forall j \in H$ yields a search plan with an objective function value that is 134 kg greater. This improved plan is computed in 11.4 hours and compromises the ability to do re-planning between mission cycles because the searchers do not return home at regular 24-hour intervals. In principle, completely relaxing the home station time window constraints so that $\tau_j^{min} = \tau_j^{start} = 0, \forall j \in H$ and $\tau_j^{max} = 72, \forall j \in H$ could result in a higher-valued search plan, however, the ability to do re-planning would be sacrificed and the computation time required to compute this plan exceeds the reasonable amount of time threshold. Lastly, as illustrated in Figures 4.17-

Figure 4.16: **CS heuristic solution to BS day 3** - High-value target SP1 is searched by both searchers as its track is closest to the home station. Searcher 2 searches SP2 and then searches GF3. Response circles for interdictor 2 relative to searcher 2 not shown.

4.20, the dwell time of searchers against targets assigned in the optimal plan is not limited by the response range of interdiction assets.

Figure 4.17 depicts the optimal search and interdiction plan in day 1 of the scenario. We see that both searchers are assigned to search GF1(3) followed by GF5(1). Interdictor 2 is positioned to support search against GF1 (and also SP1 in day 2), while interdictor 4 is assigned to support search against GF5. In contrast with the CS heuristic solution, dwell time against targets searched in day 1 of the optimal plan is limited by each searcher's endurance limit, and not by the response range of either interdictor.

Figure 4.18 depicts the optimal search and interdiction plan in day 2 of the scenario. We see that both searchers are assigned to search the high-value target SP1. This differs from the CS heuristic solution, where SP1 is searched in day 3. Interdictor 2 is positioned to support search against SP1 (and also GF1 in day 1). Again, we see that dwell time against targets searched in day 2 of the optimal plan is not limited by the response range of the interdictor. It is, however, restricted by each searcher's endurance limit and the

Searcher	Segment	t	a	d	$\boldsymbol{\theta}^+$	$\boldsymbol{\theta}^-$
	Home h_0	-	0	5.61	(89.1W 13.4N)	(89.1W 13.4N)
	GF1(3)	1.45	7.06	1.18	(82.0W 10.0N)	(82.3W 10.9N)
	GF5(1)	2.02	10.25	2.70	(83.3W 0.0N)	(85.5W 0.5S)
	Home h_1	2.66	15.61	22.39	(89.1W 13.4N)	(89.1W 13.4N)
1	SP1	1.97	39.97	6.34	(78.8W 10.7N)	(80.1W 11.6N)
	Home h_2	1.70	48.00	14.00	(89.1W 13.4N)	(89.1W 13.4N)
	GF3	1.26	63.26	4.18	(82.9W 16.3N)	(85.6W 18.5N)
	SP2(2)	1.87	69.32	1.95	(89.3W 9.0N)	(89.5W 9.5N)
	Home h_3	0.74	72.00	-	(89.1W 13.4N)	(89.1W 13.4N)
	Home h_0	-	0	5.61	(89.1W 13.4N)	(89.1W 13.4N)
	GF1(3)	1.45	7.06	1.18	(82.0W 10.0N)	(82.3W 10.9N)
	GF5(1)	2.02	10.25	2.70	(83.3W 0.0N)	(85.5W 0.5S)
	Home h_1	2.66	15.61	22.39	(89.1W 13.4N)	(89.1W 13.4N)
2	SP1	1.97	39.97	6.34	(78.8W 10.7N)	(80.1W 11.6N)
	Home h_2	1.70	48.00	14.00	(89.1W 13.4N)	(89.1W 13.4N)
	GF4(3)	0.85	62.85	3.60	(86.7W 9.5N)	(88.9W 11.5N)
	SP2(2)	0.55	67.00	4.26	(89.0W 8.5N)	(89.5W 9.5N)
	Home h_3	0.74	72.00	-	(89.1W 13.4N)	(89.1W 13.4N)

Table 4.27: **Optimal search plan for BS** - Both searchers are assigned to the same targets through the first two days of the scenario. High-value target SP1 is searched by both searchers as late as possible in the second day. This allows GF4 to enter the plan on the third day, a trade-off that was missed by all heuristic solutions.

Interdictor	Position
1	(85.6W 17.4N)
2	(80.4W 11.2N)
3	(89.4W 9.7N)
4	(84.9W 0.7S)

Table 4.28: **Optimal interdictor solution to BS** - Interdictor 1 is positioned to support search against GF3 in day 3. Interdictor 2 is positioned to support search against GF1 in day 1 and SP1 in day 2. Interdictor 3 is positioned to support search against both GF4 and SP2 in day 3. Interdictor 4 is positioned to support search against GF5 in day 1.

latest departure time from the home station $\tau_{h_1}^{max} = 38$.

Figures 4.19 and 4.20 depict the optimal search and interdiction plan in day 3 of the scenario. Since interdictor 3 is assigned to support search against two targets and both searchers for one of the two targets, we illustrate the day 3 plan for the searchers separately. In Figure 4.19 we see that the first searcher is assigned to search GF3 followed by SP2(2). Interdictor 1 is positioned to support search against GF3, while interdictor 3 is assigned to support search against SP2 (and also GF4 for the other searcher). The first searcher's

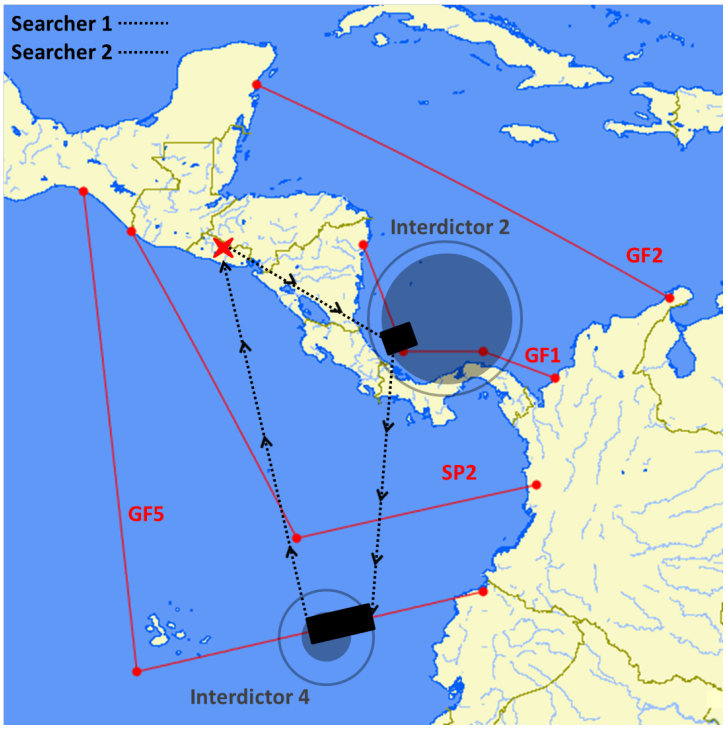


Figure 4.17: **Optimal solution to BS day 1** - Both searchers are assigned to search GF1(3) followed by GF5(1). Interdictor 2 is positioned to support search against GF1 (and also SP1 in day 2), while interdictor 4 is assigned to support search against GF5. Dwell time is limited by each searcher's endurance limit, and not by the response range of either interdictor.

dwell time in day 3 is limited by its endurance limit and the scenario time limit $D = 72$, but not the response range of either interdictor.

In Figure 4.20 we see that the second searcher is assigned to search GF4(3) followed by SP2(2). Interdictor 3 is assigned to support search against both targets (and also SP2 for the other searcher). The second searcher's dwell time in day 3 is limited by its endurance limit and the scenario time limit $D = 72$, but not the range of either interdictor.

Table 4.29 summarizes **SSP-C** solution results for our benchmark scenario **BS**. The utility of multi-cycle planning is clear. Search plans generated using either the SSP heuristic or the PBB heuristic to solve **SSP-C** have significantly higher objective function value than using a myopic approach and can be obtained within the two-hour time limit. The CS heuristic produces the highest valued search and interdiction plan within a two-hour planning time limit. If greater computing power and/or a more efficient implementation of B&B is available, obtaining the optimal search and interdiction plan to support a 24-hour

Figure 4.18: **Optimal solution to BS day 2** - Both searchers are assigned to search the high-value target SP1. Interdictor 2 is positioned to support search against SP1 (and also GF1 in day 1). Dwell time is limited by each searcher's endurance limit and the latest departure time from the home station $\tau_{h_1}^{max} = 38$, and not by the response range of the interdictor.

planning cycle may be possible.

We conclude this chapter with a remark on solving **BS** in the case where the searchers are heterogeneous. Recall that if the searchers are heterogeneous or if they do not have the same home station, it is inappropriate to use Algorithm B&B with RPPE. When we use B&B without RPPE in an attempt to solve **BS**, it takes over 24 hours to explore 10% of the enumeration tree. Clearly, given a scenario of the same size as **BS** or larger, with heterogeneous searchers, it appears unlikely that the optimal solution can be obtained in a reasonable amount of time. Either the SSP heuristic or the CS heuristic should be used.

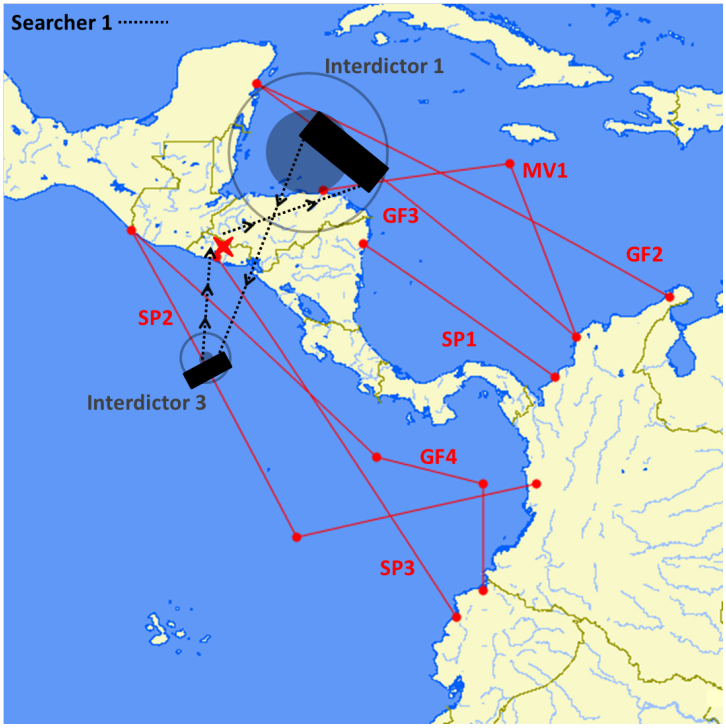


Figure 4.19: **Optimal solution to BS day 3 for searcher 1** - The first searcher is assigned to search GF3 followed by SP2(2). Interdictor 1 is positioned to support search against GF3, while interdictor 3 is assigned to support search against SP2 (and also GF4 for the other searcher). Dwell time is limited by its endurance limit and the scenario time limit $D = 72$.

Approach	Exp. Search Value (kg)	*rgap* (%)	Gap (%)	Runtime (sec)
Myopic	4469.2	53	43	480.4
SSP heuristic	6380.5	33	19	3.1
CS heuristic	7450.1	22	5	1786.3
Algorithm B&B	7851.4	-	0	24 947.9

Table 4.29: **Summary of SSP-C solution results for our benchmark scenario BS** - All three heuristic approaches produce a search and interdiction plan within the required two-hour (7,200-second) time limit. Among the heuristics, the CS heuristic yields the plan with the highest value. Objective function values are reported in expected kg of cocaine detected. The runtime required to compute the optimal solution exceeds the two-hour time limit. *rgap* and Gap refer to the relative optimality gap (4.9) and the true optimality gap respectively. Times reported do not include the 625 seconds required to obtain the upper bound of 9,588 kg.

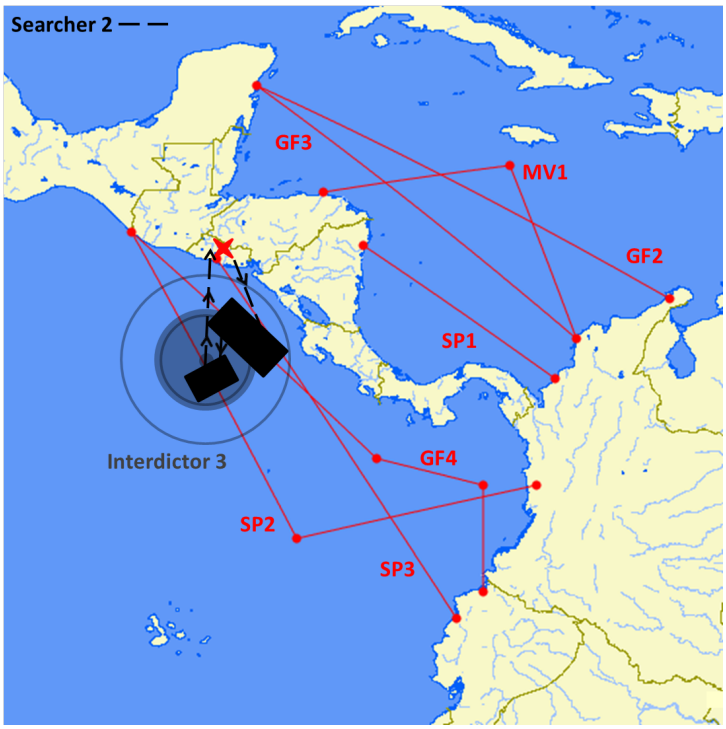


Figure 4.20: **Optimal solution to BS day 3 for searcher 2 -** The second searcher is assigned to search GF4(3) followed by SP2(2). Interdictor 3 is assigned to support search against both targets (and also SP2 for the other searcher). The second searcher's day 3 plan is limited by its endurance limit and the scenario time limit $D = 72$.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Multi-Stage Optimal Search Under Evolving Uncertainty

In general, planners will not know all scenario data with certainty. Intelligence estimates of target tracks, payload, departure and arrival information, and even sensor performance may not take on the values that planners expect. In this chapter we investigate how random planning data can impact the optimal search plans and propose an approach to handle uncertainty using a multi-stage stochastic programming model. To simplify exposition, we consider **SSP** in this study, though the findings and multi-stage stochastic programming model easily generalize to the enhanced models described in Chapter 4.

We recall that **SSP** takes as input the planning data listed in Tables 3.1 and 3.2. We study how changes to this data can affect optimal solutions. Data that is outside the control of the search controller is of primary concern. For this reason, we exclude $V, \hat{V}$, T from consideration is this analysis. While search aircraft performance may affected by external factors (e.g., weather), it is common practice to assign planning factors based on conservative aircraft performance estimates. We also exclude the scenario time limit D because this value is set at the discretion of the search controller. Lastly, we exclude the number of targets n, because removing any target j may be modeled by driving its detection value q_j to zero. Eliminating these data from consideration and allowing sweep width W take a different value for each target j, we arrive at the *scenario data* listed in Table 5.1.

Searcher sensor sweep width against target j	W_j
Speed of target j	U_j
Expected departure time of target j	τ_j
Time uncertainty range of target j	$\tilde{\tau}_j$
Expected departure location of target j	$\boldsymbol{\rho}_j$
Expected arrival location of target j	$\bar{\boldsymbol{\rho}}_j$
Departure/arrival location uncertainty range of target j	$\tilde{\rho}_j$
Expected value of detecting target j	q_j

Table 5.1: **Scenario data**

We assume that scenario data are random with known probabilistically distributions, and that their distributions have finite support. For any discrete random variable $\tilde{\omega}$ we denote

by $P(\omega)$ the probability that $\tilde{\omega} = \omega$.

5.1 Sensitivity of Optimal Search Plans

Given a fixed scenario ω (i.e., data in Table 5.1), we wish to solve **SSP** and determine how its optimal solution is affected by perturbations to ω. We denote by ω' a perturbed scenario which is created by changing one or more of the scenario data elements that make up ω. One of three things can happen: (1) The optimal solution in scenario ω may still be optimal in scenario ω'. (2) The optimal solution in scenario ω may be infeasible in scenario ω'. (3) The optimal solution in scenario ω may be suboptimal in scenario ω'. The first possibility requires no investigation.

The second possibility is caused by changes to **SSP** constraints. For example, if the time windows, e.g., constraints (3.2f) and (3.2g), for some target j were to shift so this target is active before another target i, where $\tau_i^{min} > \tau_j^{max}$, then $x_{i,j} = 1$ would be infeasible. If arc (i, j) were in the optimal path $\boldsymbol{x}^*$ in scenario ω, this previously optimal solution would no longer be feasible in scenario ω'. While a change of this type appears to be problematic, since perturbations are known probabilistically, we account for alternative target tracks by adding artificial targets to the model, adjusting the expected search value q of each target to reflect its probability of occurrence. For this reason, we are more concerned with changes to scenario data that result in suboptimal search plans.

The third possibility is driven by changes to the **SSP** objective function (3.2a). The objective function depends on the expected detection value q and the detection rate α. Since the objective function is linear in q, when the detection value is a random variable $\tilde{\boldsymbol{q}}$ with a set of possible values $\mathcal{Q}$, the objective function using the distribution for $\tilde{\boldsymbol{q}}$ is equivalent to the objective function using its expected value. More precisely, given objective function $f(\boldsymbol{d}, \boldsymbol{q}) = \boldsymbol{q}\hat{f}(\boldsymbol{d})$, the following is true.

$$E_{\tilde{\boldsymbol{q}}}[f(\boldsymbol{d}, \tilde{\boldsymbol{q}})] = \sum_{\boldsymbol{q} \in \mathcal{Q}} P(\boldsymbol{q}) f(\boldsymbol{d}, \boldsymbol{q}) = \sum_{\boldsymbol{q} \in \mathcal{Q}} P(\boldsymbol{q}) \boldsymbol{q} \hat{f}(\boldsymbol{d}) = E_{\tilde{\boldsymbol{q}}}[\tilde{\boldsymbol{q}}] \hat{f}(\boldsymbol{d})$$

The same cannot be said about α because (3.2a) is a nonlinear function of the detection rate. Recall that the detection rate α (3.1) is a function of the searcher's on-station speed $\hat{V}$, the sensor sweep width W, and the search region area $\tilde{\tau}_j \tilde{\rho}_j U_j$. Since the searcher has control over its speed, changes to $\hat{V}$ are of little concern. Changes to the search

region area are not considered here because they are more appropriately captured by adding alternative target tracks. The focus of this sensitivity analysis is on examining how changes to sweep width W can affect the optimal search plan.

It is well known (see United States Coast Guard 2013, appx. H) that environmental conditions impact sensor performance. When a visual sensor is used, sweep width can diminish due to any number of factors (e.g., cloud cover, wind, etc.). Alternatively, when a radar sensor is used, sweep width can be affected by environmental clutter and noise, and atmospheric conditions. We now proceed to analyze how changes in sensor performance impact optimal search plans.

5.1.1 Search Allocation Problem

We begin by considering a search allocation problem (SAP). The SAP is a distribution of search effort problem (see Stone 1975, ch. 1) that arises when transit time $\boldsymbol{t} = \mathbf{0}$ is possible and time window constraints (3.2f) and (3.2g) are relaxed in **SSP**. Examining this problem provides insight into how the **SSP** is impacted by changes in W. The SAP is stated as follows.

Problem SAP

$$\max_{\boldsymbol{d}} \quad \sum_{j \in \hat{N}} q_j \left(1 - \exp\left\{-\alpha_j d_j\right\}\right) \tag{5.1a}$$

$$\text{s.t.} \quad \sum_{j \in \hat{N}} d_j \leq T \tag{5.1b}$$

$$d_j \geq 0, \quad \forall j \in \hat{N} \tag{5.1c}$$

Allowing sweep width to depend on each target $j \in \hat{N}$, we introduce the partial detection rate $\hat{\alpha}$, which excludes sweep width.

$$\hat{\alpha}_j = \frac{\hat{V}}{\tilde{\tau}_j \tilde{\rho}_j U_j} = \frac{\alpha_j}{W_j} \tag{5.2}$$

Since **SAP** is a convex NLP, the first order KKT necessary conditions (see Bertsekas 1999, ch. 3) are also sufficient. Denoting by $\hat{N}^*$ the set of inactive constraints (5.1c)

corresponding to positive dwell times, the KKT conditions are reduced to express the optimal dwell times $\boldsymbol{d}^*$ as follows.

$$d_j^* = \frac{1}{W_j\hat{\alpha}_j \sum\limits_{i\in\hat{N}^*} 1/(W_i\hat{\alpha}_i)} \left(T - \sum_{j'\in\hat{N}^*} \frac{1}{W_{j'}\hat{\alpha}_{j'}} \log\{W_{j'}\hat{\alpha}_{j'}q_{j'}\} \right) + \frac{1}{W_j\hat{\alpha}_j} \log\{W_j\hat{\alpha}_j q_j\} \quad (5.3)$$

We observe that (5.3) expresses each d_j^* as a function of itself and the other search dwell times by virtue of $\hat{N}^*$. Optimal dwell times are obtained by a simple node-addition procedure (Stone 1975, ch. 2) where $\hat{N}^* = \emptyset$ initially, then targets $i \in \hat{N}\backslash\hat{N}^*$ corresponding to the largest value $W_i\hat{\alpha}_i q_i$ are added to $\hat{N}^*$ in turn until adding another would violate constraint (5.1c). Note that this procedure inversely corresponds to the *Remove_j* procedure considered in step 8 of the SSP heuristic described in Section 3.2.1. The natural interpretation of these procedures is to add to (or remove from) the search plan the targets with the largest (or smallest) rate-of-reward.

Though it is not obvious from (5.3), increasing W_j may increase or decrease d_j^* depending on the values of $T, q_j, \hat{\alpha}_j, W_i, q_i$, and $\hat{\alpha}_i$ for the other targets $i \in \hat{N}\backslash\{j\}$. Generally speaking, if d_j^* is currently "small", then increasing W_j will increase d_j^*. However, if d_j^* is currently "large", then increasing W_j will decrease d_j^*. Naturally, "small" and "large" are relative, so we proceed to consider **SAP** with two targets to demonstrate this effect.

5.1.2 Two-Target Analysis

Given **SAP** where $|\hat{N}| = 2$, the optimal solution (5.3) reduces to the following.

$$d_1^* = \min\left\{T, \max\left\{0, \frac{W_2\hat{\alpha}_2 T + \log\{W_1\hat{\alpha}_1 q_1\} - \log\{W_2\hat{\alpha}_2 q_2\}}{W_1\hat{\alpha}_1 + W_2\hat{\alpha}_2}\right\}\right\} \quad (5.4)$$

$$d_2^* = T - d_1^* \quad (5.5)$$

If we assume T and $\hat{\alpha}$ take on values listed in Table 5.2, and W can vary within the range $[1, 20]$, we arrive at the contoured sweep-sweep plot in Figure 5.1.

The contoured sweep-sweep plot depicts the proportion of the searcher endurance T that is allocated to target 1 as the sweep width for each target varies independently. The

W	$[1, 20]$
T	10
$\hat{\alpha}$	0.0137

Table 5.2: **Two-target analysis data** - W varies independently for each target. $\hat{\alpha}$ is the same for both targets. $\hat{\alpha}$=0.0137 results from setting $\hat{V} = 205$ and $\tilde{\tau}_j \tilde{\rho}_j U_j = 15,000$.

white region of the plot is where all of the searcher endurance is allocated to target 1, while the dark blue region is where all of the searcher endurance is allocated to target 2. The contours fading from white to dark blue indicate a mixture of the searcher endurance allocation between both targets. Contours closer to the white region favor target 1 in the allocation mixture, while contours closer to the dark blue region favor target 2.

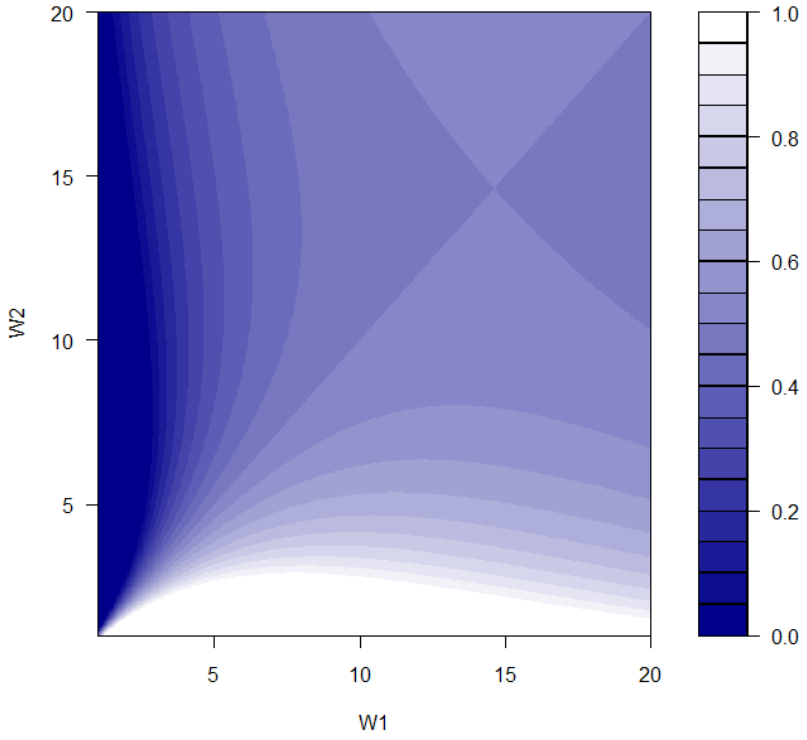
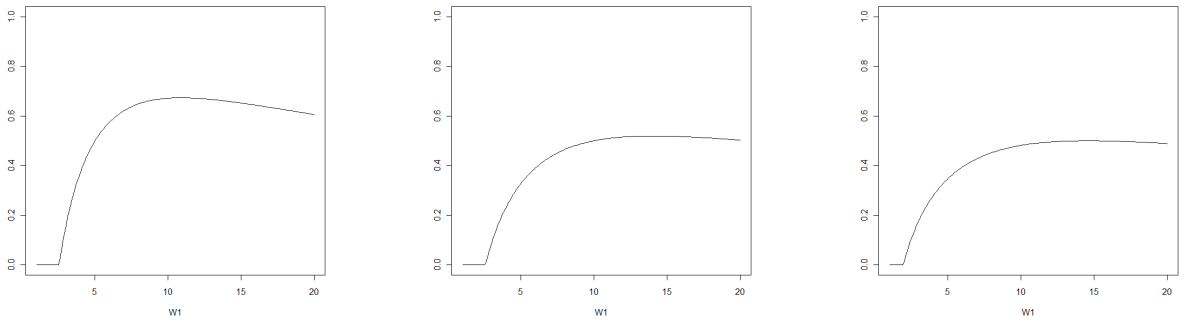


Figure 5.1: **SAP contoured sweep-sweep plot** - The white region indicates where all of the searcher endurance is allocated to target 1, while the dark blue region is where all of the searcher endurance is allocated to target 2. The contours fading from white to dark blue indicate a mixture of the searcher endurance allocation between both targets. Contours closer to the white region favor target 1 in the allocation mixture, while contours closer to the dark blue region favor target 2.

Figure 5.2 depicts the proportion of the searcher endurance T that is allocated to target 1 as the sweep width against target 1 varies for fixed target 2 sweep widths $W_2 \in \{5, 10, 15\}$. The three line plots shown are created by taking horizontal slices of the contoured sweep-sweep plot Figure 5.1.



(a) (b) (c)

Figure 5.2: **Optimal target 1 dwell time allocation as a function of W_1 for fixed W_2** - (a) $W_2 = 5$. (b) $W_2 = 10$. (c) $W_2 = 15$. Changes to W_1 have a greater impact in increasing optimal target 1 dwell time when both W_1 and W_2 are small. The effect of increasing W_1 is less pronounced when either W_1 or W_2 is large.

Figures 5.1 and 5.2 highlight several features of **SAP** solutions that would reasonably apply to **SSP** solutions as well. First, we see that increasing sweep width against a particular target does not necessary result in greater dwell time against this target in the optimal solution. Secondly, when the sweep width against one target is fixed and not too small, search dwell time for the other target goes to zero as sweep width against it decreases. This feature highlights the need to take a close look at the solution *switch points* that mark the boundary between where search dwell time for a target is zero and where it is nonzero.

The switch-point sweep-sweep plot in Figure 5.3 depicts the three regions whose boundaries correspond to where search dwell time for a target switches from zero to nonzero, and vice versa. The light blue region of the plot is where all of the searcher endurance is allocated to target 1, while the dark blue region is where all of the searcher endurance is allocated to target 2. The white region indicates a mixture of the searcher endurance allocation between both targets. We refer to this region as the *mixture region* of the switch-point sweep-sweep plot. If transit time $\boldsymbol{t} = \mathbf{0}$ were possible in **SSP**, these three

regions (light blue, dark blue, and white) would correspond to differing values of the optimal path variable $\boldsymbol{x}^*$. Clearly if dwell time against a target is zero, it cannot be optimal to route a searcher to this target's search region.

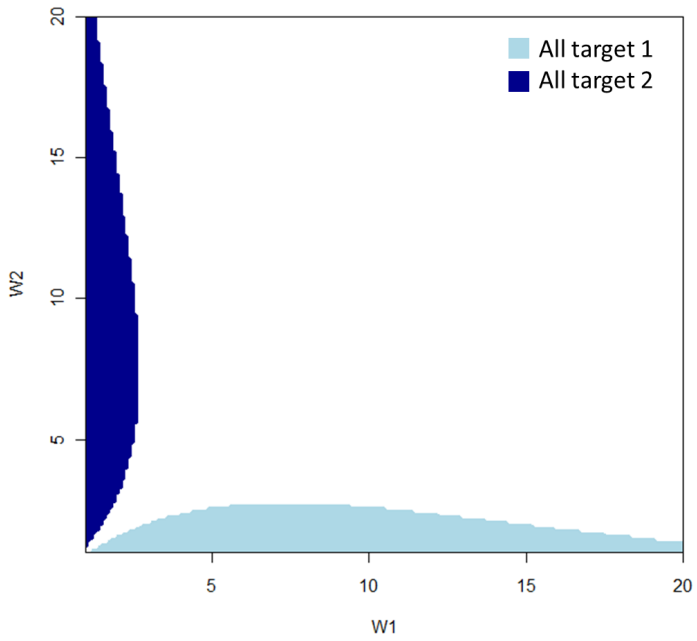


Figure 5.3: **SAP sweep-sweep plot without contours** - The light-shaded (blue) region indicates where all of the searcher endurance is allocated to target 1, while the dark-shaded (blue) region is where none of the searcher endurance is allocated to target 1.

In **SSP**, where $\boldsymbol{t} = \mathbf{0}$ transit time is not possible, the searcher is not able to use all of its endurance for search dwell time $\boldsymbol{d}$. In **SAP** this has the affect of reducing T. Figure 5.4 depicts the switch-point sweep-sweep plot for **SAP** with $T = 5$. We see that the mixture region is smaller than that of Figure 5.3 when the searcher endurance is twice as large. The mixture region is even smaller when we consider the optimal allocation of searcher endurance $T = 2$ shown in Figure 5.5. We should expect the optimal solution to **SSP** to behave similarly. Since the searcher must transit from one search region to the next, the result of this transit is diminished opportunity to search all targets. In scenarios where a large amount of transit time is required for the searcher to execute the search plan, we should expect that the optimal solution is most sensitive to changes in sweep width because the switch points are close together.

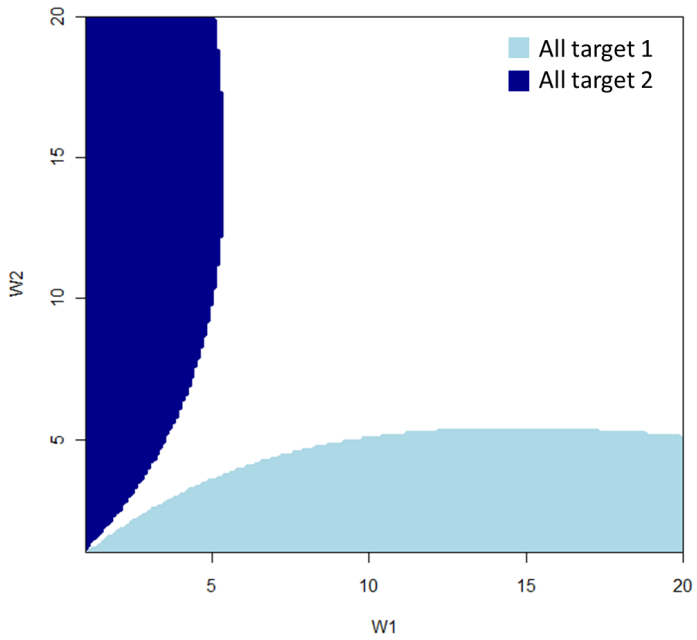


Figure 5.4: **SAP sweep-sweep plot without contours with reduced searcher endurance** - When the searcher endurance is reduced to $T = 5$ hours, the region (white) where both targets are searched in the optimal solution is smaller than that of Figure 5.3 when the searcher endurance is twice as large.

5.1.3 SSP Sweep Width Experiment

We now consider **SSP** applied to two different two-target scenarios in order to illustrate how the impact of changes to sweep width depends on the transit time that is required for the searcher to execute the search plan. The first scenario, scenario 1, considers targets that are relatively close together in time and space. In this scenario, a small amount of transit time is required for the searcher to search both targets. The second scenario, scenario 2, considers targets that are relatively far apart in time and space. In this scenario, substantial transit time is required for the searcher to search both targets. Targets 2 and 3 are considered in scenario 1, while targets 1 and 4 are considered in scenario 2. Data for both scenarios are given in Tables 5.3 and 5.4. We allow sweep width W to vary within the range $[1, 20]$ for each target independently. Figure 5.6 depicts a spatial representation of the target tracks.

Computing the optimal solution to **SSP** in scenario 1 for varying sweep width values, we obtain the switch-point sweep-sweep plot in Figure 5.7. Given that targets 2 and 3

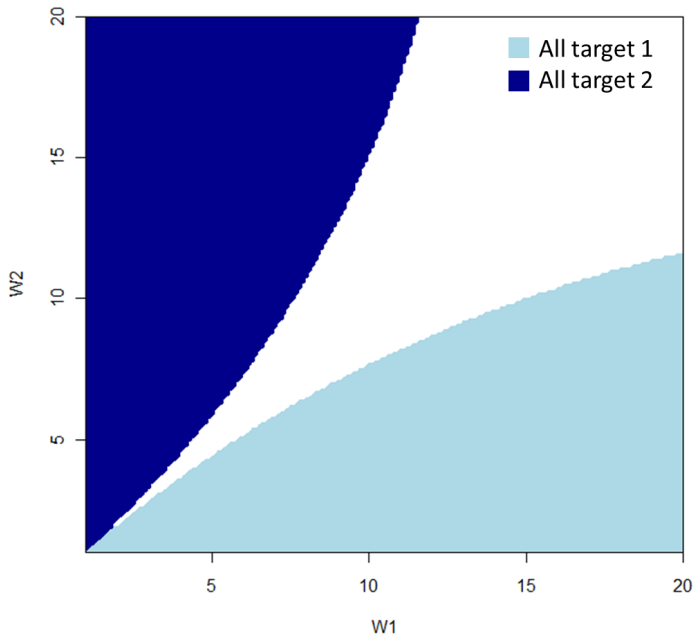


Figure 5.5: **SAP sweep-sweep plot without contours with reduced searcher endurance** - When the searcher endurance is reduced to $T = 2$ hours, the mixture region (white) where both targets are searched in the optimal solution is small. In this situation, the optimal solution is highly sensitive to changes in sweep width.

Home	(85.5W 10.6N)
V	325
$\hat{V}$	205
T	10
D	24

Table 5.3: **Searcher data for sweep width experiments**

are close together, a small amount of time is required for the searcher to transit from one search region to another. As a result, as evidenced by the large mixture region in this figure, the optimal allocation of the searcher endurance between the targets is, in a relative sense, not very sensitive to changes in sweep width. Results in this scenario resemble the **SAP** analysis with searcher endurance $T = 10$ illustrated in Figure 5.3.

Computing the optimal solution to **SSP** in scenario 2 for varying sweep width values, we obtain the switch-point sweep-sweep plot in Figure 5.8. Given that targets 1 and 4 are far apart, a large amount of time is required for the searcher to transit from one search

	Target			
	1	2	3	4
τ	0	0	0	10
$\boldsymbol{\rho}$	(81W 1S)	(80W 0N)	(79W 1N)	(76.3W 9N)
$\bar{\boldsymbol{\rho}}$	(96.5W 15.7N)	(92W 14.5N)	(91W 14N)	(83.5W 14N)

Table 5.4: **Target data derived by the search controller** - Additionally, $\tilde{\tau} = 3$, $\tilde{\rho} = 100$, $q = 1000$, and $U = 50$ for all four targets. Scenario 1 includes targets 2 and 3. Scenario 2 includes targets 1 and 4.

Figure 5.6: **Target track map for SSP sweep width experiments** - Scenario 1 includes targets 2 and 3. Scenario 2 includes targets 1 and 4.

region to another. As a result, as evidenced by the small mixture region in this figure, optimal allocation of the searcher endurance between the targets is, relative to scenario 1, more sensitive to changes in sweep width. Results in this scenario resemble the **SAP** analysis with searcher endurance $T = 5$ illustrated in Figure 5.4.

Note that, in scenario 1, for all sweep width configurations W_1, W_2 within the mixture region of Figure 5.7, the optimal searcher paths $\boldsymbol{x}^*$ are the same. Provided both targets are searched, the optimal search order is always target 3 followed by target 2. The same is true in scenario 2. Provided both targets are searched, the optimal search order is always

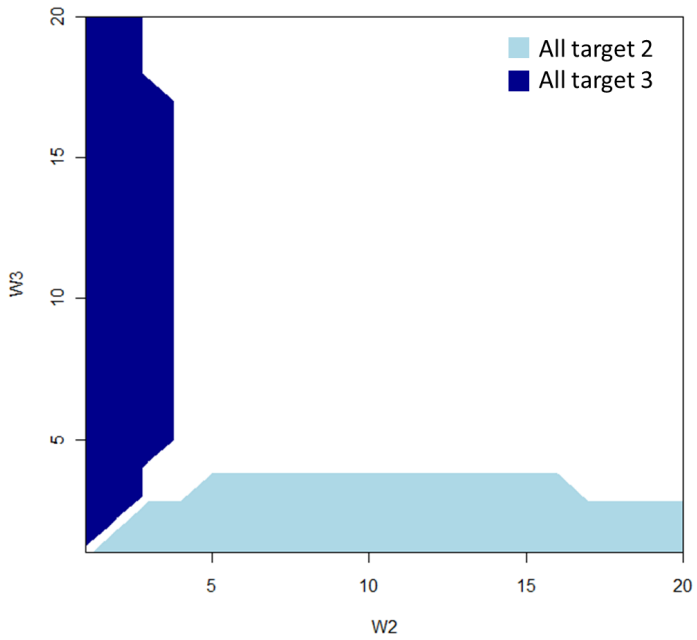


Figure 5.7: **Switch-point sweep-sweep plot for scenario 1** - Given that targets targets 2 and 3 are close together, little time is required for the searcher to transit from one search region to another. Optimal allocation of the searcher endurance between the targets is not very sensitive to changes in sweep width.

target 1 followed by target 4.

This is noteworthy because if it is true in general, for an SSP with n targets, that there exists an underlying optimal search order where all targets can be ordered independent of sweep width, then accounting for changes in sweep width could be done efficiently. For example, suppose that we are interested in an SSP with 10 targets, we have baseline scenario data that provides values $W_j, j = 1, 2, \ldots, 10$, and we have computed the underlying optimal search order. If any of the sweep widths W_j were to change, then re-optimizing the search plan would simply entail determining which targets should be searched and which targets should not. The search order would not change. In terms of **SSP**, it would be sufficient to solve the problem with the binary visitation variables $\boldsymbol{y}$ instead of the binary path variables $\boldsymbol{x}$. Since $\boldsymbol{y} \in \{0,1\}^{n+2}$ and $\boldsymbol{x} \in \{0,1\}^{n^2+2(n+1)}$, re-optimizing with the modified sweep width data could be done more efficiently than completely re-solving **SSP**. A simple modification of branching rules in step 2 of Algorithm B&B to preserve the search order can be made to achieve this efficiency.

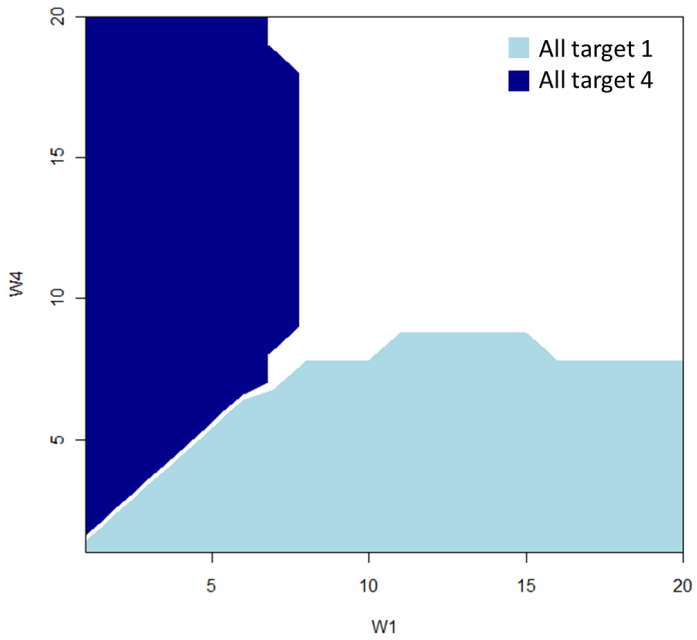


Figure 5.8: **Switch-point sweep-sweep plot for scenario 2** - Given that targets targets 1 and 4 are far apart, more is required for the searcher to transit from one search region to another. Optimal allocation of the searcher endurance between the targets is more sensitive to changes in sweep width.

Unfortunately, an underlying optimal search order does not exist in general. If we consider a 3-target SSP with targets 2, 3, and 4 from Table 5.4, we can see that the optimal search order changes with changing sweep width values. For most sweep width configurations W_2, W_3, W_4, target 3 is searched before target 2. This order agrees with our results for scenario 1. However, for sweep width configurations where it is optimal to search all three targets (e.g., $W_2 = 10, W_3 = 10, W_4 = 15$), the optimal search order is target 2, target 3, and then target 4. The order of targets 2 and 3 in the optimal search path is reversed.

As a result of this sensitivity analysis we see that changes to sweep width can completely change the optimal search plan. Targets can exit the optimal search plan or enter the optimal search plan when a switch-point is crossed. The relative search order of targets can also change as switch-points are crossed and, more troublingly, even within the interior of the mixture region where some combination of targets are searched. In any search planning situation, planners need to consider the certainty level associated with scenario data, be aware of how changes to this data may affect the plan, and, if possible, directly

account for data uncertainty in the model. We now address the latter.

5.2 Multi-Stage Planning

We consider daily mission cycles where planning (or re-planning) occurs L hours prior to the start of each mission execution period. At each planning epoch, given the available scenario data, a search plan is computed for the next mission execution period and beyond. When scenario data only extends up to the next day, we have a single-cycle problem that can be modeled by **SSP** with appropriate model enhancements described in Chapter 4. On the other hand, when scenario data extends beyond the next 24 hours, we have a multi-cycle problem that can be modeled by **SSP-C** with appropriate model enhancements. A timeline for this mission planning process is shown in Figure 5.9.

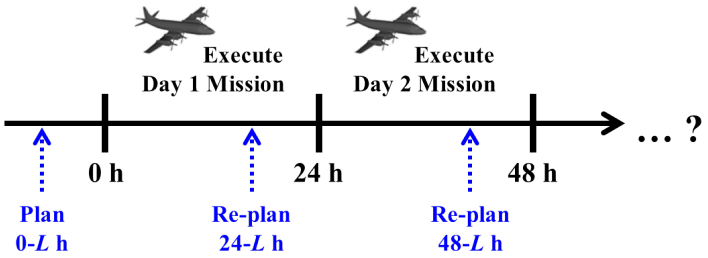


Figure 5.9: **Mission planning timeline** - Planning (or re-planning) occurs L hours prior to each mission execution period. At each planning epoch, given the available scenario data, a search plan is computed for the next mission execution period and beyond.

In this mission planning process, we need to consider three things that can happen between planning epochs. (1) New target information can arrive. (2) Updated weather forecasts can become available. (3) Targets can be detected. The latter is not addressed by our models. When a target is detected, it necessarily changes the scenario data for subsequent mission cycles. A positive detection implies that the target was detected somewhere along the path of the searcher. In this way, the scenario data for later mission cycles would depend on the solution for earlier mission cycles. A dynamic programming model (see, e.g., Kress et al. 2012, Lehnerdt 1982) is more appropriate to handle this consideration.

The first two considerations are direct changes to the scenario data and may occur independent of search activity. Assuming that the new data can be modeled as a random vector with finite support, we can account for data uncertainty with a multi-stage stochastic program (see, e.g., King and Wallace 2012, Watson and Woodruff 2010, Shapiro et al. 2009, Rockafellar and Wets 1991). A multi-stage model represents all possible scenario

data realizations by a number of scenarios $\omega \in \Omega$, each with a probability of occurrence $P(\omega)$. We refer to this data as *probabilistic scenario data.*

As a practical matter, in order to use this model, when the initial plan is made at time $0 - L$, planners must be able to estimate possible updates to scenario data that will arrive leading up to subsequent re-planning epochs, along with the probability that each update will be realized. The nature of the real-world search missions makes this requirement unrealistic for some types of scenario data. In most cases, it is unreasonable to think that planners will be able to include information that has to do with new target information in the probabilistic scenario data. Intelligence indicating the existence of a future target is more appropriately captured in the model in the first place. However, planners would likely be able to estimate possible updates to scenario data that are due to weather. Weather models often characterize future conditions with probabilities of occurrence (Gombos et al. 2012, Hansen et al. 2011). Figure 5.10 is a rendering of graphic which shows how, in an abstract scenario, sensor performance may be affected by environmental conditions (Hansen 2013).

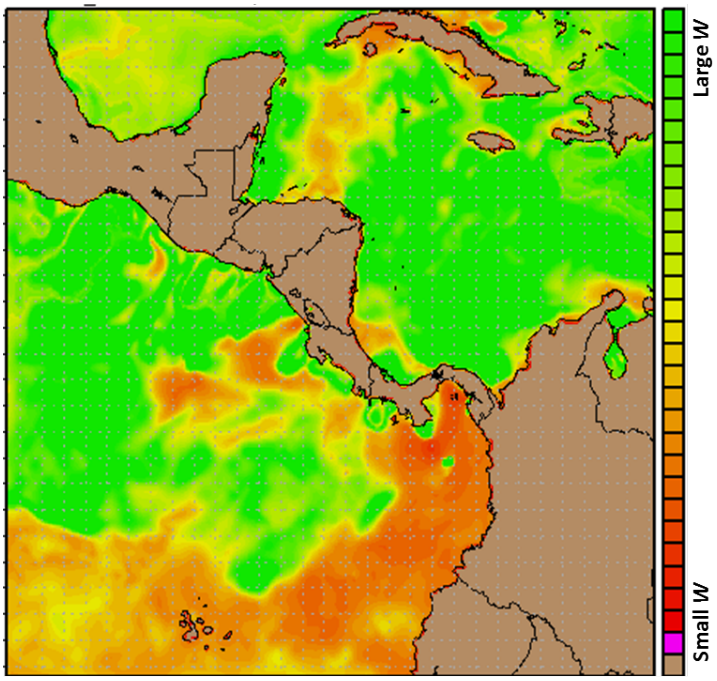


Figure 5.10: **Spatial distribution of sensor performance** - Red-orange areas reflect small sweep width (poor sensor performance) corresponding to bad weather. Green areas reflect large sweep width (high sensor performance) corresponding to bad weather.

We assume that nominal start times for intermediate home stations $\tau_j^{start}, \forall j \in \hat{H}$ corre-

spond to daily mission cycles as described in Section 4.6. For each mission cycle $k \in C$, we define the set scenario bundles $\hat{\Omega}_k \subseteq \{(\omega, \omega') : \omega \in \Omega, \omega' \in \Omega\}$, where scenarios ω and ω' are indistinguishable up to and including cycle k. As an example, since the day 1 scenario data is known prior to initial planning at time $0 - L$, all scenarios are equivalent in the first mission cycle. All scenarios are in the first bundle $\hat{\Omega}_1 = \Omega \times \Omega$. In later cycles $k > 1$, bundles are sparser because the scenarios are dissimilar. Indexing all **SSP-C** variables $\boldsymbol{a}, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{x}, \boldsymbol{y}$, and $\boldsymbol{z}$ on scenarios $\omega \in \Omega$, and using a scenario-expanded network $G_\Omega = (N_\Omega, A_\Omega)$, which is constructed equivalently as the searcher-expanded network G_S in Section 4.3, we arrive at the following multi-stage SSP.

Problem SSP-MSP:

$$\max_{\boldsymbol{a},\boldsymbol{d},\boldsymbol{t},\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \quad \sum_{\omega\in\Omega} P(\omega)\left(\sum_{j\in\hat{N}_\Omega} q_j^\omega\left(1-\exp\Big\{-\alpha_j^\omega d_j^\omega y_j^\omega\Big\}\right)\right) \tag{5.6a}$$

$$\text{s.t.} \qquad a_j^\omega z_{k,j} - a_j^{\omega'} z_{k,j} = 0, \qquad \forall k \in C, j \in N_\Omega, \; (\omega,\omega') \in \hat{\Omega}_k \tag{5.6b}$$

$$d_j^\omega z_{k,j} - d_j^{\omega'} z_{k,j} = 0, \qquad \forall k \in C, j \in N_\Omega, \; (\omega,\omega') \in \hat{\Omega}_k \tag{5.6c}$$

$$t_{i,j}^\omega z_{k,i} z_{k,j} - t_{i,j}^{\omega'} z_{k,i} z_{k,j} = 0, \qquad \forall k \in C, (i,j) \in A_\Omega, \; (\omega,\omega') \in \hat{\Omega}_k \tag{5.6d}$$

$$x_{i,j}^\omega z_{k,i} z_{k,j} - x_{i,j}^{\omega'} z_{k,i} z_{k,j} = 0, \qquad \forall k \in C, (i,j) \in A_\Omega, \; (\omega,\omega') \in \hat{\Omega}_k \tag{5.6e}$$

(3.2b), (3.2c), (3.2e) – (3.2m), and (4.5b) – (4.5h), $\qquad \forall \omega \in \Omega$

The objective function (5.6a) calculates the expected value of the search using the probability of occurrence for each scenario. Constraints (5.6b)-(5.6e) enforce non-anticipativity for each scenario bundle. Constraints (3.2b), (3.2c), (3.2e)-(3.2m), and (4.5b)-(4.5h) are enforced for each scenario $\omega \in \Omega$.

Problem **SSP-MSP** is solved using Algorithm B&B on the scenario-expanded network.

We observe, however, that this model becomes too large to solve in a reasonable amount of time when a large number scenarios are considered. The number of partial paths that must be considered in B&B grows exponentially with the number of scenarios. A popular heuristic that solves multi-stage stochastic programming problems with many scenarios is the progressive hedging algorithm (Rockafellar and Wets 1991, Watson and Woodruff 2010). Progressive hedging is discussed further in Section 5.2.2. We now proceed with a numerical example solved using Algorithm B&B.

5.2.1 SSP-MSP Example

In order to illustrate the utility of **SSP-MSP**, we consider a planning problem with a single searcher, four targets, and two 24-hour mission cycles. The planning data for this problem is given in Tables 5.5 and 5.6.

Home	(89.1W 13.4N)
V	325
$\hat{V}$	205
T	10
D	24
T_R	6

Table 5.5: **Searcher data for SSP-MSP example**

	Target			
	1	2	3	4
q	4000	3000	3000	4000
τ	14	0	0	36
$\boldsymbol{\rho}$	(80W 0N)	(78.7W 2N)	(76.3W 9N)	(72W 12N)
$\bar{\boldsymbol{\rho}}$	(92.2W 14.5N)	(85.5W 10N)	(83.5W 14N)	(87.5W 20N)

Table 5.6: **Target data for SSP-MSP example** - Additionally, $\tilde{\tau} = 3$, $\tilde{\rho} = 100$, and $U_j = 50$ for all four targets.

Figure 5.11 depicts a spatial representation of the target tracks for this planning problem. A high value target, target 1, is active at the end of day 1, and due to its relatively long movement track, it remains active through the first part of day 2. Targets 2 and 3 are active early in the first day of the scenario and have relatively short movement tracks. Another high value target, target 4, has a long movement track and is active late in day 2. Due to the timing of target movement tracks, searching for target 1 in day 1 cannot be done effectively if either target 2 or target 3 is also searched in day 1. Similarly, searching for target 1 in day 2 cannot be done effectively if target 4 is also searched in day 2.

Figure 5.11: **Spatial representation of the target tracks for SSP-MSP example** - Target 1 is active at the end of day 1, and due to its relatively long movement track, it remains active through the first part of day 2. Target 1 is active at the end of day 1, and due to its relatively long movement track, it remains active through the first part of day 2. Targets 2 and 3 are active early in the first day of the scenario and have relatively short movement tracks. Target 4, with a long movement track, is active late in day 2.

We assume that the sensor performance distribution depicted in Figure 5.10 illustrates the general weather trend in day 2. Given this, we expect that bad weather (i.e., small sweep width) is more likely than good weather (i.e., large sweep width) in the Eastern Pacific region. Similarly, we expect that good weather (i.e., large sweep width) is more likely than bad weather (i.e., small sweep width) in the Caribbean region. We assume good weather in both regions in day 1. For the remainder of this chapter, we denote by $j(k)$ target j searched in day k. Variables and parameters are indexed in this fashion when appropriate. For example, we denote by $W_{j(k)}$ the sweep width against target j in day k. The probabilistic scenario data for this problem is given in Table 5.7.

This example highlights a difficult issue that planners may face: a *here-and-now* versus *wait-and-see* trade-off. A single searcher cannot effectively search for all targets in two mission cycles. A planner must choose between going after a high value target, target 1, in day 1 at the expense of neglecting targets 2 and 3, or saving this high value target for day

	Scenario			
	ω_1	ω_2	ω_3	ω_4
Probability	0.1875	0.0625	0.5625	0.1875
E. Pacific Weather	Good	Good	Bad	Bad
Caribbean Weather	Good	Bad	Good	Bad
$W_{j(1)}, \forall j \in \hat{N}$	15	15	15	15
$W_{1(2)}, W_{2(2)}$	15	15	5	5
$W_{3(2)}, W_{4(2)}$	15	5	15	5

Table 5.7: **Probabilistic scenario data for SSP-MSP example** - Good weather (i.e., large sweep width) is expected throughout the AOI in the first day. The second day is marked by bad weather and small sweep width in the Eastern Pacific region, and good weather and large sweep width in the Caribbean region.

2 while running the risk of a bad weather scenario. In classical stochastic programming terms, day 1 requires here-and-now decisions, which must be made before the scenario is revealed. In contrast, day 2 requires wait-and-see decisions, which can be made after the scenario is revealed (Wets 2002, Shapiro et al. 2009).

A *gambling planner* guesses which scenario will be realized, perhaps choosing the most likely scenario or, based on professional judgment, another less-likely scenario, and determines his search plan by solving **SSP-C** with a single scenario. A *hasty planner* determines his search plan by solving **SSP-C** with a single scenario as well, however he uses the expected sweep width in the model. A *cautious planner* is uneasy about the uncertainty associated with the weather and would like to have a plan that he can adjust based on the day 2 weather outcome, so he uses **SSP-MSP** to compute his plan. It is not obvious what solutions these approaches would yield. We now proceed to compare them.

For the remainder of this chapter, the following convention is used. We denote by π search plan. Three plans, π_1, π_2, and π_3, are central to the understanding of the approaches taken by the the gambling planner, the hasty planner, and the cautious planner. Table 5.8 lists the optimal dwell times and searcher path for each plan.

We distinguish between three measures of search plan effectiveness. The optimal objective function value for a particular problem is denoted by Z^*. The expected search plan value given a search plan π to be executed and a scenario ω realization is denoted by *value*. The expected search plan value across all scenarios given a search plan π to be executed is denoted by $E[value]$. Note that the objective function (5.6a) maximizes $E[value]$ over all

Plan	$d^*_{1(1)}$	$d^*_{1(2)}$	$d^*_{2(1)}$	$d^*_{3(1)}$	$d^*_{4(2)}$	Searcher Path
π_1	5.67	0	0	0	5.65	$h_0 \rightarrow 1(1) \rightarrow h_1 \rightarrow 4(2) \rightarrow h_2$
π_2	0	8.5	2.89	3.06	0	$h_0 \rightarrow 3(1) \rightarrow 2(1) \rightarrow h_1 \rightarrow 1(2) \rightarrow h_2$
π_3	0	0	2.89	3.06	5.65	$h_0 \rightarrow 3(1) \rightarrow 2(1) \rightarrow h_1 \rightarrow 4(2) \rightarrow h_2$

Table 5.8: **Search plan details for SSP-MSP example** - Data in this table reflects the optimal dwell times $\boldsymbol{d}^*$ for a given path $\boldsymbol{x}$. The Searcher Path column corresponds to the path obtained by sequencing the $x_{i,j} = 1$ values of the associated path $\boldsymbol{x}$ for the associated plan π. Plans π_1, π_2, and π_3 correspond to optimal search plans for the different planning approaches examined in this section.

possible search plans π, and this is equivalent to Z^* only when **SSP-MSP** is the problem that is being solved. The *value* for each search plan π and each scenario ω is given in Table 5.9.

	Scenario Realized				
Plan	ω_1	ω_2	ω_3	ω_4	$E[value]$
π_1	5492.8	4030.1	5492.8	4030.1	5127.2
π_2	6038.7	6038.7	4501.3	4501.3	4885.6
π_3	5482.9	4020.1	5482.9	4020.1	5117.2

Table 5.9: **Expected search plan value by plan for SSP-MSP example** - Data in this table reflect the *value* for each plan π_o given that scenario ω is realized. $E[value]$ is computed for each scenario using probabilities in Table 5.7.

The gambling planner assumes he "knows" the scenario outcome and determines the optimal search plan accordingly. This approach entails choosing the appropriate column in Table 5.7 which corresponds to the desired scenario, and solving **SSP-C** with this scenario data. A gambling planner who assumes that scenario ω_3 will be realized computes π_1 as his optimal plan. A gambling planner who assumes that any other scenario will be realized computes π_2. Given that ω_3 occurs with probability 0.5625, π_1 appears to be a favorable choice for a gambling planner.

The hasty planner uses the expected sweep width within **SSP-C** to compute his search plan. He calculates that, on day 2, the sweep width in the Eastern Pacific and Caribbean regions will be 7.5 nm and 12.5 nm respectively. Using these values, he computes π_1, with an objective function value $Z^* = 5,227.0$ kg, as his optimal plan.

The cautious planner chooses to use **SSP-MSP** to compute his search plan. With this approach, the optimal policy is to choose π_3 if scenario ω_3 is realized, and choose π_2 if any other scenario is realized. We refer to this scenario-based contingency plan as π_0. The $E[value]$ of π_0 is 5,437.8 kg, which is greater than that of the other plans π_1, π_2, and

π_3. Figure 5.12 illustrates that π_0 yields the largest worst case *value* in any scenario. We observe that π_0 is implementable because non-anticipativity constraints (5.6b)-(5.6e) are satisfied; the day 1 portions of π_2 and π_3 are identical. Planners who choose π_0 make the here-and-now versus wait-and-see trade-off by setting a day 1 plan at time $0 - L$, while retaining the flexibility to adjust the day 2 plan at planning epoch $24 - L$ when the weather outcome is known.

Figure 5.12 illustrates how the *value* of two plans, π_0 and π_1, taken by the three planners compare for each scenario realization. In three of the four scenarios, the cautious planner who chooses π_0 does substantially better than the hasty and the gambling planners who choose π_1. When ω_3 is realized, the hasty and the gambling planners do slightly better by choosing π_1, with a *value* that is 10 kg greater than the plan π_0 chosen by the cautious planner.

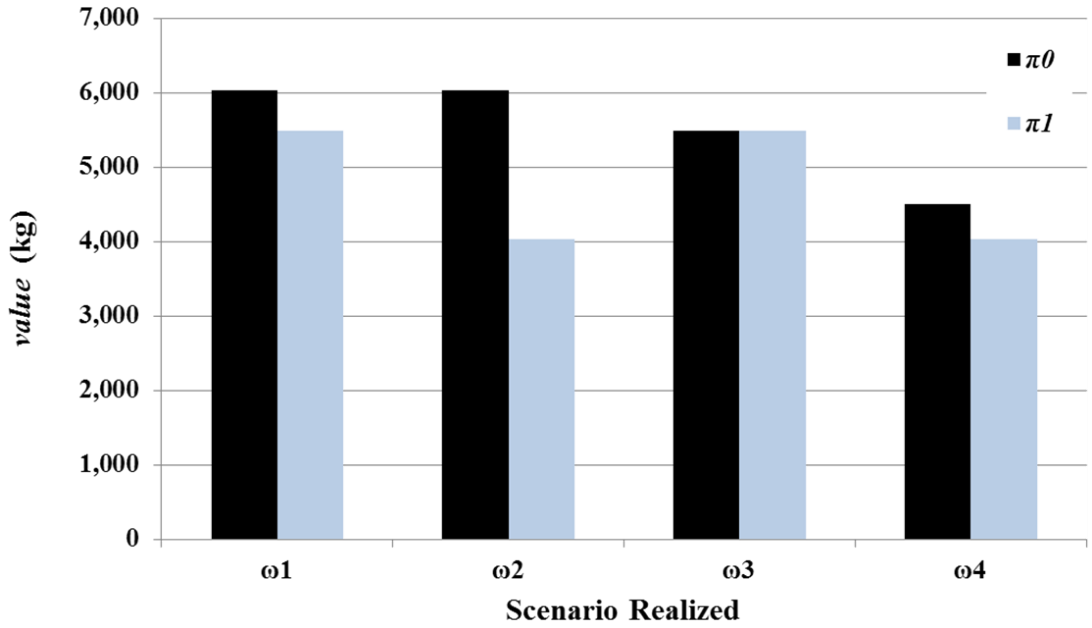


Figure 5.12: **Comparison of planning approaches** - In three of the four scenarios, the cautious planner who chooses π_0 does substantially better than the hasty and the gambling planners who choose π_1. When ω_3 is realized, the hasty and the gambling planners do slightly better by choosing π_1, with a *value* that is 10 kg greater than the plan π_0 chosen by the cautious planner.

Given that scenario ω_3 is the most likely scenario and that π_1 is optimal if expected sweep width is used, it would be plausible for a planner to choose this search plan in the absence of multi-stage planning. Unfortunately for this planner, with probability 0.4375, his plan will be inferior to the scenario-based contingency plan π_0. Clearly, π_0 is the most desirable

option, yielding the largest $E[value]$, the largest worst case *value*, and a *value* that is only 10 kg less than that of π_1 in scenario ω_3.

5.2.2 Progressive Hedging Algorithm

Since real-world mission planning with many scenarios yields large problems that may preclude computing solutions to **SSP-MSP** using Algorithm B&B, planners wishing to do multi-stage planning may have to rely on heuristics such as the progressive hedging algorithm (PH). PH computes solutions by relaxing the non-anticipativity constraints (5.6b)-(5.6e), and then iterating between solving the decoupled problems **SSP-C** for each scenario ω and penalizing non-anticipativity constraint violation. PH takes two parameters: a penalty parameter $\boldsymbol{\rho}^{\mathrm{PH}}$ and a stopping tolerance ϵ^{PH}. The parameter $\boldsymbol{\rho}^{\mathrm{PH}}$ penalizes violations of non-anticipativity constraints within the algorithm, while the stopping tolerance ϵ^{PH} indicates the degree to which non-anticipativity constraint violations are acceptable. The complete PH algorithm can be found in Watson and Woodruff (2010, p. 2).

PH is an exact algorithm for convex optimization problems (Rockafellar and Wets 1991), while it is a heuristic for problems with integer variables, like **SSP-MSP**. In multi-stage planning for real-world search missions, we are most concerned with evaluating multiple scenarios in a time-constrained environment. While computing the optimal solution to **SSP-MSP** is ideal, if a heuristic can compute a solution in a reasonable amount of time, and this solution is better than the solution obtained by solving **SSP-C** using expected (sweep width) value, then this heuristic can be a valuable tool for planners.

In order to evaluate the performance of PH on large problems, which B&B is not expected to be able to solve in a reasonable amount of time, we consider an excursion to the example problem of Section 5.2.1 where we vary the number of scenarios. Scenarios are constructed as follows. We assume that the sweep width against target 1 in day 2 is a random variable $\tilde{W}_{1(2)}$ with a scaled beta distribution.

$$\frac{\tilde{W}_{1(2)}}{20} \sim beta(\alpha = 2.4, \beta = 4)$$

We assume that the sweep width against target 4 in day 2 is a random variable $\tilde{W}_4$,

independent of $\tilde{W}_{1(2)}$, with a scaled beta distribution.

$$\frac{\tilde{W}_{4(2)}}{20} \sim beta(\alpha = 4, \beta = 2.4)$$

The particular distributions chosen in this excursion are arbitrary and carry no real-world significance in sweep width uncertainty. Choosing underlying sweep width distributions in this excursion allows us to attribute differences in algorithm performance to changes in the number of scenarios. We eliminate the potential for algorithms to perform differently due to changes to the underlying probability distributions. These distributions yield sweep widths in the range $(0, 20)$ nm and have identical expected sweep widths as those used by the hasty planner in Section 5.2.1. We denote by EV the approach taken by the hasty planner - using expected sweep width to compute the search plan. We consider problems with 4, 6, 9, 10, 15, 25, and 100 scenarios by discretizing the joint distribution into bins according to sweep width levels. Since the sweep width values in each problem have the same underlying distribution, the expected sweep widths are all the same and we can compare the $E[value]$ in each problem to that of the same baseline plan π_1 (see Table 5.8) computed using EV. This allows us to assess the quality of PH solutions when a large number of scenarios precludes solving the problem to optimality using Algorithm B&B. We use PH with parameter values $\rho_j^{\mathrm{PH}} = q_j, \forall j \in N_\Omega$; $\rho_{i,j}^{\mathrm{PH}} = \max\{q_i, q_j\}, \forall (i,j) \in A_\Omega$; and $\epsilon^{\mathrm{PH}} = 1e-3$. We require that, in order for solutions to be useful to planners, they be computed in a reasonable amount of time. We define a reasonable amount of time to be two hours.

Table 5.10 lists the $E[value]$ for the plans computed by B&B, PH, and EV as the number of scenarios increases. Note that the differences in $E[value]$ for EV are due to probability binning when discretizing the joint sweep width probability distribution. EV yields the same plan π_1 and objective function value $Z^* = 5,227.0$ kg in all cases. For problems with 4,6, and 9 scenarios PH correctly identifies the optimal solution as a scenario-based contingency plan made up of π_2 and π_3. Recall that π_0 is a scenario-based contingency plan made up of π_2 and π_3, where four scenarios are used. For problems with more scenarios, the PH solution is also a scenario-based contingency plan made up of π_2 and π_3. In all cases, the PH solution has a greater $E[value]$ than the EV solution.

Figure 5.13 depicts the runtimes of B&B,PH, and EV on a log scale as the number of scenarios is increased. Runtimes for B&B, as expected, grow exponentially with the number

	Scenario Realized		
Scenarios	B&B (optimal)	PH	EV
4	5437.8	5437.8	5127.2
6	5408.9	5408.9	5127.2
9	5364.1	5364.1	5116.6
10	-	5423.2	5127.2
15	-	5386.8	5116.6
25	-	5469.5	5212.4
100	-	5386.6	5163.5

Table 5.10: **Expected search plan values for increasing number of scenarios** - Data in this table reflect the $E[value]$ for each solution method for increasing number of scenarios. For problems with 4,6, and 9 scenarios PH correctly identifies the optimal solution as a scenario-based contingency plan made up of π_2 and π_3. For problems with more scenarios, the PH solution is a scenario-based contingency plan made up of π_2 and π_3 as well. In all cases, the PH solution has a greater $E[value]$ than the EV solution.

of scenarios. With nine scenarios, B&B requires over 15 hours to compute the optimal solution. Using B&B to solve **SSP-MSP** appears to be a poor option for planners in a time constrained environment if they would like to consider any more than six scenarios. Runtimes for PH appear grow linearly with the number of scenarios. A simple linear regression fit of the PH runtimes versus number of scenarios yields a 6.7-second-per-scenario runtime increase rate (e.g., slope), a slope p-value which is less than 10^{-9}, and an R^2 value of 0.9998. All PH solutions are computed within two hours. With 100 scenarios, PH requires 11 minutes to compute a solution that yields a larger $E[value]$ than the EV-based plan π_1. PH appears to be a useful tool for planners who would like to consider multi-stage planning with many scenarios and do so in a reasonable amount of time.

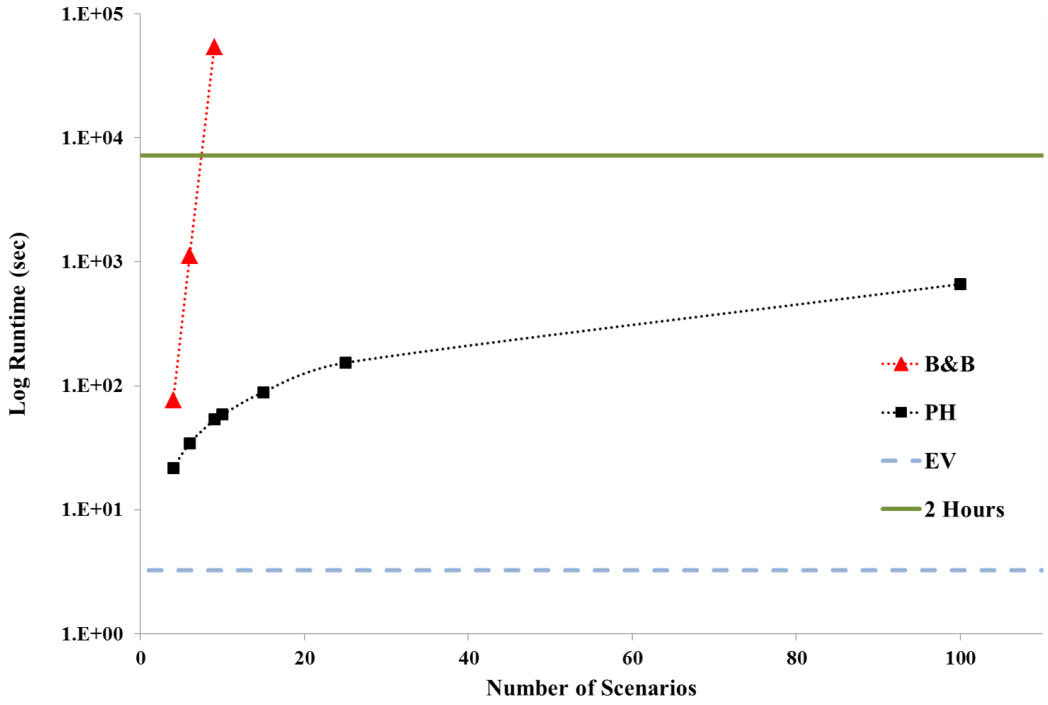


Figure 5.13: **Runtime comparison for multi-stage example** - Runtimes for B&B, as expected, grow exponentially with the number of scenarios. With nine scenarios, B&B requires over 15 hours to compute the optimal solution. Runtimes for PH are all below two hours. With 100 scenarios, PH requires 11 minutes to compute a solution that yields a larger $E[value]$ than the EV-based plan π_1. The runtime for EV is constant at 3.25 seconds for any number of scenarios.

CHAPTER 6:
Summary, Conclusions, and Future Work

6.1 Summary and Conclusions

This research presents the Generalized Orienteering Problem with Resource Dependent Rewards (GOP-RDR), which appears to be the first Orienteering Problem (OP) to consider generalizations of the node rewards and arc lengths at the same time. We show that in the GOP-RDR the activity of collecting rewards at nodes is in direct competition with the activity of transiting between nodes. We develop a specialized branch-and-bound (B&B) algorithm, which exploits this competition for resources. The GOP-RDR and B&B can be used to model and solve problems in many application areas, such as mission planning for search aircraft, commercial vehicle routing, sports, tourism, production, and scheduling.

We present the Smuggler Search Problem (SSP), a novel path-constrained optimal search model in continuous time and space, as an important special case of the GOP-RDR. The SSP is used to compute an optimal search plan for a single searcher that is routed in an area of interest (AOI) to detect multiple linearly moving targets. The SSP requires many fewer integer variables than classical discrete-time and -space optimal search models. We present a specialized B&B algorithm and three heuristics tailored to this model. We show that these algorithms are able to quickly compute optimal search plans in scenarios that are on the scale of real-world counterdrug operations, a first in the field of optimal search. Numerical results show that B&B applied to randomly generated SSP instances with seven or more targets outperforms standard mixed-integer nonlinear programming solvers. We present a search order heuristic that can be used to quickly compute good search plans, with runtimes within one second for problems of up to 10 targets. We demonstrate empirically that optimality gaps, when using the search order heuristic, tend to remain stable, in the 1-3% range, as problem size is increased from 3 to 10 targets.

We study five enhancements to the SSP, along with tailored solution procedures, that can be used together or separately to model various real-world search planning scenarios. With these enhancements, planners can account for complex target motion, multi-vehicle search planning, high uncertainty in target motion, the coordination of search and in-

terdiction efforts, and multi-period search planning over time. The SSP, combined with these enhancements, is the first optimal search model to consider all of these issues. The merits of each SSP enhancement are demonstrated on a benchmark scenario, which is developed to highlight several issues that may be faced by real-world search planners.

In the first SSP enhancement, we account for complex target motion by approximating each target's movement path with a piecewise-linear track of segments. This model may be used in planning scenarios where the target's speed and/or direction changes along its path, as well as when the searcher's performance characteristics vary in the AOI (e.g., due to weather conditions). We note that approximating a target's path by adding segments is equivalent to adding nodes in the network G considered in the SSP. We find that this model, with seven target segments and a single searcher, can be solved to optimality using B&B in 3.5 seconds.

In the second SSP enhancement, we account for multiple cooperating, heterogeneous searchers. We solve problem instances of this type using B&B on a searcher-expanded network. Since the enumeration tree grows exponentially with the number of searchers, B&B should only be used solve problem instances with a moderate number of searchers. In our experiment with the baseline scenario, we observe that optimal multi-searcher plans often assign more than one searcher to the same target. This is especially true when the scenario involves "high value" targets. We find that this model, with seven target segments and two searchers, can be solved to optimality using B&B in 48.9 seconds, while heuristic solutions, with optimality gaps less than 10%, can be computed in as little as 1.3 seconds.

With the third SSP enhancement, we demonstrate that scenarios with high uncertainty in a target's movement track may be modeled by allowing search in fixed regions. This model allows planners to account for targets whose departure time uncertainty value is larger than the duration of the mission execution period. We find that, though fixed-region targets in the baseline scenario are not assigned to be searched in the optimal plan, this model may be useful when fixed-region targets have a relatively high search value. We demonstrate that this model, with eight target segments and two searchers, can be solved to optimality using B&B in 64.8 seconds, while heuristic solutions, with optimality gaps less than 10%, can be computed in as little as 1.4 seconds.

In the fourth SSP enhancement, we coordinate aerial search efforts with the positioning of surface interdictors. This is an important consideration in counterdrug operations

where searchers must rely on surface assets to physically intercept smugglers once they are detected by the searcher. By adding continuous variables and second order cone range constraints to the search model, we develop the first search-and-interdiction model that can be used for planning in real-world counterdrug operations. Based on our study of this model, we recommend that it be used to compute improved search plan recommendations whenever interdiction support is required. We find that this model, with eight target segments, two searchers, and four interdictors, can be solved to optimality using B&B in 84.8 seconds, while heuristic solutions, with optimality gaps less than 10%, can be computed in as little as 1.5 seconds.

The fifth SSP enhancement is a multi-period search model that accounts for sequencing search plans over multiple mission execution cycles. We show that this multi-period model may be used to improve plans when the search needs to be coordinated with the positioning of interdictors over several mission execution periods. We find that this model, with 16 target segments, two searchers, four interdictors, and three mission execution cycles, can be solved using B&B in 6.9 hours. Runtimes as short as 3.1 seconds can be achieved by heuristics, which yield solutions with search plan values that are within 19% of the optimal plan. Based on our numerical experiments with this model, we recommend that, when solutions are required in a time constrained environment, heuristics be used to solve problem instances with two or more heterogeneous searchers and three or more mission execution cycles.

We study the sensitivity of optimal search plans with respect to weather uncertainty. Based on this study, we conclude that planners need to consider the certainty level associated with scenario data, be aware of how changes to this data may affect the plan, and, if possible, directly account for data uncertainty by using a multi-stage search planning model.

We present a multi-stage model that allows planners to account for uncertainty in scenario data. This appears to be the first multi-stage optimal search model that is able to handle problems on the scale of those encountered in real-world counterdrug operations. We observe that multi-stage planning, as opposed to planning for a single scenario, can yield higher quality search plans with reduced risk of poor outcomes due to weather uncertainty. We solve problem instances of this type using B&B on a scenario-expanded network. We find that the multi-stage model, with five target segments, a single searcher, and four

scenarios, can be solved to optimality using B&B in 1.5 hours. Since the enumeration tree grows exponentially with the number of scenarios, we recommend that B&B only be used solve problem instances with a moderate number of scenarios. In order to solve larger problem instances, we show that the progressive hedging algorithm may be used to solve multi-stage problems with many scenarios. We find that the multi-stage model, with five target segments, a single searcher, and up to 100 scenarios, can be solved using progressive hedging in within 1.5 hours.

6.2 Future Work

A common approach in optimal search problems is to represent intelligence about target motion and environmental forecasts with a discrete-time and -space distribution of target presence. The models presented in this dissertation are formulated in continuous time and space, assuming piecewise linear target motion, so that the resulting optimization problem can be solved by planners in a time-constrained environment. It is not clear how to best translate discrete-time and -space distributions of target presence into appropriate inputs to a continuous-time and -space model. It seems reasonable to use linear regression techniques to approximate target motion, however establishing a complete approximation procedure requires further research.

The SSP heuristic, based on results presented in Section 3.3, appears to perform well on single-searcher problems with one mission cycle. We present modifications to the SSP heuristic which account for multiple searchers and multiple mission cycles. It seems reasonable to improve these modified heuristics by expanding on the clustering concept used in the single-searcher, single-cycle SSP heuristic. Observing that the SSP strives to make dwell times large and transit times small, clustering targets based on searcher and/or mission cycle assignment may yield higher quality solutions.

The SSP uses a random search model to maximize the expected value of the search. Maximizing the expected number of targets detected can be done by simply setting equal to one the expected detection value for each target. Other objectives may be of interest to planners. For example, some optimal search problem formulations aim to maximize the probability that at least one target is detected (see, e.g., Foraker 2011). Alternatively, a different search model (e.g., exhaustive search) may be considered.

Search aircraft not only have endurance limits that restrict the duration of single missions,

but they also have aggregate flying hour limits that constrain their total usage over long periods of time (e.g., monthly or annual flying hour programs). This type of "long-range" constraint is not considered in this dissertation. It seems reasonable to consider a multi-cycle model that extends many time periods into the future in order to assess search plan feasibility with respect to long-range usage constraints. Such a consideration brings two significant challenges. First, efficient algorithms would be required to solve the multi-cycle problem over many periods. Second, one would need to find the right balance between projecting potentially inaccurate target intelligence data too far into the future versus accounting for long-range constraints on search aircraft.

This dissertation presents numerical results for search problem scenarios with up to 16 target segments, two searchers, and three mission execution cycles. Algorithm B&B runtimes for larger problem instances are greater than 24 hours. Alternative algorithms need to be developed in order to solve larger problem instances to optimality within a reasonable amount of time.

The models presented in this dissertation were developed to support maritime search for drug smugglers. A natural extension appears to be to consider land-based search missions. In some ways, a land-based search mission appears to be easier to model because target tracks may be limited to existing roadways. A piecewise linear motion model, like **SSP-PWL** in Section 4.2, may be used in the case when targets can move freely over land. More study seems warranted to develop an appropriate search value model to use in the objective function; it is not clear whether or not random search of a moving search region is valid when considering land-based search missions.

THIS PAGE INTENTIONALLY LEFT BLANK

REFERENCES

Alpern, S., S. Gal. 2003. *The Theory of Search Games and Rendezvous*. International Series in Operations Research & Management Science, Kluwer, New York, NY.

Archetti, C., A. Hertz, M. Speranza. 2007. Metaheuristics for the team orienteering problem. *Journal of Heuristics* **13**(1) 49–76.

Benkoski, S., M. Monticino, J. Weisinger. 1991. A survey of search theory literature. *Naval Research Logistics* **38** 469–494.

Bertsekas, D. P. 1999. *Nonlinear Programming,* 2nd ed. Athena Scientific, Belmont, MA.

Boussier, S., D. Feillet, M. Gendreau. 2007. An exact algorithm for team orienteering problems. *4OR A Quarterly Journal of Operations Research* **5**(3) 211–230.

Butt, S., T. M. Cavalier. 1994. A heuristic for the multiple tour maximum collection problem. *Computers and Operations Research* **21**(1) 101–111.

Butt, S., D. Ryan. 1999. An optimal solution procedure for the multiple tour maximum collection problem using column generation. *Computers and Operations Research* **26** 427–441.

Camm, J. D., A. S. Raturi, S. Tsubakitani. 1990. Cutting Big M down to size. *Interfaces* **20** 61–66.

Chao, I., B. Golden, E. Wasil. 1996a. A fast and effictive heuristic for the orienteering problem. *European Journal of Operational Research* **88**(3) 475–489.

Chao, I., B. Golden, E. Wasil. 1996b. The team orienteering problem. *European Journal of Operational Research* **88**(3) 464–474.

Dell, R., J. Eagle, G. Martins, A. Santos. 1996. Using multiple searchers in constrained-path, moving-target search problems. *Naval Research Logistics* **43**(4) 463–480.

Desrochers, M., G. Laporte. 1991. Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. *Operations Research Letters* **10** 27–36.

Dillenberger, C. 1994. On practical resource allocation for production planning and scheduling with period overlapping setups. *European Journal Of Operational Research* **75**(2) 275–286.

Dolan, E. D., J. J. Moré. 2002. Benchmarking optimization software with performance profiles. *Mathematical Programming* **91**(2) 201–213.

Eagle, J., J. Yee. 1990. An optimal branch-and-bound procedure for the constrained path, moving target search problem. *Operations Research* **38**(1) 110–114.

Erdogan, G., G. Laporte. 2013. The orienteering problem with variable profits. *Networks* **61**(2) 104–116.

Erkut, E., J. Zhang. 1996. The maximum collection problem with time-dependent rewards. *Naval Research Logistics* **43**(5) 749–763.

Escudero, L. F., J. Salmeron. 2005. On a fix-and-relax framework for a class of project. *Annals of Operations Research* **140**(1992) 163–188.

Evers, L., T. Dollevoet, A. I. Barros, H. Monsuur. 2012. Robust UAV mission planning. *Annals of Operations Research* doi:10.1007/s10479-012-1261-8.

Foraker, J. 2011. Optimal search for moving targets in continuous time and space using consistent approximations. Dissertation, Naval Postgraduate School, Monterey, CA.

Gendreau, M., G. Laporte, F. Semet. 1998. A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks* **32**(4) 263–273.

Gill, P. G., W. Murray, M. Saunders, A. Drud. 2013. SNOPT user manual. Retrieved June 10, 2013, http://www.gams.com/dd/docs/solvers/snopt.pdf.

GlobalSecurity.org. 2013. Reliable security information. Retrieved June 10, 2013, http://www.globalsecurity.org/index.html.

Golden, B., L. Levy, R. Vohra. 1987. The orienteering problem. *Naval Research Logistics* **34**(3) 307–318.

Golden, B., S. Raghavan, E. Wasil. 2008. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, New York, NY.

Gombos, D., R. N. Hoffman, J. A. Hansen. 2012. Ensemble statistics for diagnosing dynamics: Tropical cyclone track forecast sensitivities revealed by ensemble regression. *Monthly Weather Review* **140** 2647–2669.

Grossmann, I. E., J. Viswanathan, A. Vecchiette, R. Raman, E. Kalvelagen. 2013. DICOPT user manual. Retrieved March 11, 2013, http://www.gams.com/dd/docs/solvers/dicopt.pdf.

Grundel, D. 2005. Constrained search for a moving target. *Proceedings of the 2005 international conference on Collaborative technologies and systems*. CTS'05, IEEE Computer Society, Washington, DC, St. Louis, MO, 327–332.

Hansen, J. A. 2013. COAMPS 12-km forecasts project: sCentAM EM sensor performance.

Hansen, J. A., G. Jacobs, L. Hsu, J. Dykes, J. Dastugue, R. Allard, C. Brown, D. Lalejini, M. Abramson, S. Russell, R. Mittu. 2011. Information domination: Dynamically coupling METOC and INTEL for improved guidance for piracy interdiction. *2011 Naval Research Lab Review* 88–98.

Ilhan, T., S. Iravani, M. Daskin. 2008. The orienteering problem with stochastic profits. *IIE Transactions* **40**(4) 406–421.

Jin, Y., Y. Liao, A. A. Minai, M. M. Polycarpou. 2006. Balancing search and target response in cooperative unmanned aerial vehicle (UAV) teams. *IEEE transactions on systems man and cybernetics Part B Cybernetics a publication of the IEEE Systems Man and Cybernetics Society* **36**(3) 571–587.

Joint Interagency Task Force South. 2013. Serving the nation for over 20 years. Retrieved March 20, 2013, http://www.jiatfs.southcom.mil/index.aspx.

King, A. J., S. W. Wallace. 2012. *Modeling With Stochastic Programming*. Operations Research and Financial Engineering, Springer, New York, NY.

Kress, M., J. O. Royset, N. Rozen. 2012. The eye and the fist: Optimizing search and interdiction. *European Journal of Operational Research* **220**(2) 550–558.

Kumar, S. N., R. Panneerselvam. 2012. A survey on the vehicle routing problem and its variants. *Intelligent Information Management* **4**(3) 66–74. doi:10.4236/iim.2012.43010.

Laporte, G., S. Martello. 1990. The selective traveling salesman problem. *Discrete Applied Mathematics* **26** 193–207.

Lehnerdt, M. 1982. On the structure of discrete sequential search problems and of their solutions. *Mathematische Operationsforschung und Statistik Series Optimization* **13**(4) 523–557.

Moser, H. D. 1990. Scheduling and routing tactical aerial reconaissance vehicles.

Munsing, E., C. J. Lamb. 2011. *Joint Interagency Task Force-South: The Best Known, Least Understood Interagency Success*. Strategic Perspectives, National Defense University Press, Washington, D.C.

Murtagh, B. A., M. A. Saunders, P. A. Gill. 2013. MINOS user manual. Retrieved March 11, 2013, http://www.gams.com/dd/docs/solvers/minos.pdf.

Pietz, J., J. O. Royset. 2013. Generalized orienteering problem with resource dependent rewards. *Naval Research Logistics* **60**(4) 294–312.

Privé, J., J. Renaud, F. Boctor, G. Laporte. 2005. Solving a vehicle-routing problem arising in soft-drink distribution. *Journal of the Operational Research Society* **57**(9) 1045–1052.

Ramesh, R., K. M. Brown. 1991. An efficient four-phased heuristic for the generalized orienteering problem. *Computers & Operations Research* **18**(2) 151–165.

Ramesh, R., Y. Yoon, M. Karwan. 1992. An optimal algorithm for the orienteering tour problem. *ORSA Journal on Computing* **4**(2) 155–165.

Riehl, J. R., G. E. Collins, J. P. Hespanha. 2007. Cooperative graph-based model predictive search. *Decision and Control, 2007 46th IEEE Conference on*. 2998–3004. doi:10.1109/CDC.2007.4435025.

Rockafellar, R. T., R. Wets. 1991. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research* **16**(1) 119–147.

Royset, J. O., D. Reber. 2009. Optimized routing of unmanned aerial systems for the interdiction of improvised explosive devices. *Military Operations Research* **14**(4) 5–19.

Royset, J. O., H. Sato. 2010. Route optimization for multiple searchers. *Naval Research Logistics* **57**(8) 701–717.

Sahinidis, N., M. Tawarmalani. 2013. BARON user manual. Retrieved March 11, 2013, http://www.gams.com/dd/docs/solvers/baron.pdf.

Sato, H., J. O. Royset. 2010. Path optimization for the resource-constrained searcher. *Naval Research Logistics* **57**(5) 422–440.

Schilde, M., K. F. Doerner, R. F. Hartl, G. Kiechle. 2009. Metaheuristics for the bi-objective orienteering problem. *Swarm Intelligence* **3**(3) 179–201.

Shapiro, A., D. Dentcheva, A. Ruszczynski. 2009. *Lectures on stochastic programming: modeling and theory*. Society for Industrial Mathematics, Philadelphia, PA.

Silberholz, J., B. Golden. 2010. The effective application of a new approach to the generalized orienteering problem. *Journal of Heuristics* **16**(3) 393–415.

Souffriau, W., P. Vansteenwegen, G. Vanden Berghe, D. Van Oudheusden. 2013. The multiconstraint team orienteering problem with multiple time windows. *Transportation Science* **47**(1) 53–63.

Souffriau, W., P. Vansteenwegen, J. Vertommen, G. Vanden Berghe, D. Van Oudheusden. 2008. A personalized tourist trip design algorithm for mobile tourist guides. *Applied Artificial Intelligence* **22**(10) 964–985.

Spall, J. C. 2012. Cyclic seesaw process for optimization identification. *Journal of optimization theory and applications* **154** 187–208.

Stewart, T. 1979. Search for a moving target when searcher motion is restricted. *Computers and Operations Research* **6** 129–140.

Stone, L. D. 1975. *Theory of Optimal Search*. Academic Press, New York, NY.

Sun, A. K. 2009. Cooperative UAV search and intercept. Master's thesis, University of Toronto, Toronto, ON, Canada.

Tang, H., E. Miller-Hooks. 2005. A tabu search heuristic for the team orienteering problem. *Computers & Operations Research* **32**(6) 1379–1407.

Tawarmalani, M., N. Sahinidis. 2004. Global optimization of mixed integer nonlinear programs: A theoretical and computational study. *Mathematical Programming* **99** 563–591.

Toth, P., D. Vigo. 2002. *The vehicle routing problem*, *SIAM Monographs on Discrete Mathematics and Applications*, vol. 9. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Tricoire, F., M. Romauch, K. F. Doerner, R. F. Hartl. 2010. Heuristics for the multi-period orienteering problem with multiple time windows. *Computers & Operations Research* **37**(2) 351–367.

Trummel, K., J. Weisinger. 1986. The complexity of the optimal searcher path problem. *Operations Research* **34**(2) 324–327.

Tsiligirides, T. 1984. Heuristic methods applied to orienteering. *Journal of the Operational Research Society* **35**(9) 797–809.

United States Coast Guard. 2013. U. S. Coast Guard Addendum to the U. S. National Search and Rescue Supplement to the International Aeronautical and Maritime Search and Rescue Manual COMDTINST M16130.2F. Retrieved April 30, 2013, http://www.uscg.mil/hq/cg5/cg534/sar_manuals.asp.

United States Southern Command. 2013. Missions main. Retrieved June 10, 2013, http://www.southcom.mil/ourmissions/Pages/Our-Missions.aspx.

Vansteenwegen, P., W. Souffriau, D. Van Oudheusden. 2011. The orienteering problem: A survey. *European Journal of Operational Research* **209**(1) 1–10.

Vigerske, S. 2013. COIN user manual. Retrieved March 11, 2013, http://www.gams.com/dd/docs/solvers/coin.pdf.

Wang, Q., X. Sun, B. L. Golden, J. Jia. 1995. Using artificial neural networks to solve the orienteering problem. *Annals of Operations Research* **61**(1) 111–120.

Wang, X., B. Golden, E. Wasil. 2008. Using a genetic algorithm to solve the generalized orienteering problem. B. Golden, S. Ragahavan, E. Wasil, eds., *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, New York, NY, 264–274.

Washburn, A. R. 2002. *Search and Detection,* 4th ed. INFORMS, Linthicum, MD.

Watson, J., D. Woodruff. 2010. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science* **8**(4) 355–370.

Wets, R. 2002. Stochastic programming models: Wait-and-see versus here-and-now. C. Greengard, A. Ruszczynski, eds., *Decision Making under Uncertainty: Energy and Environmental Models*. Springer, New York, NY, 1–16.

Wolsey, L. A. 1997. Mip modelling of changeovers in production planning and scheduling problems. *European Journal of Operational Research* **99** 154–165.

Wong, E., F. Bourgault, T. Furukawa. 2005. Multi-vehicle bayesian search for multiple lost targets. *Proceedings of the 2005 IEEE Internationl Conference on Robotics and Automation*. Barcelona, Spain, 3169–3174.

Yang, Y., A.A. Minai, M. M. Polycarpou. 2004. Decentralized cooperative search by networked uavs in an uncertain environment. *Proceedings of the 2004 American Control Conference*, vol. 6. Boston, MA, 5558–5563.

Yi, J. 2003. Vehicle routing with time windows and time-dependent rewards: A problem from the american red cross. *Manufacturing & Service Operations Management* **5**(1) 74–77.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudly Knox Library
   Naval Postgraduate School
   Monterey, California